NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MISSION ASSIGNMENT MODEL AND SIMULATION TOOL FOR DIFFERENT TYPES OF UNMANNED AERIAL VEHICLES

by

Yücel Alver
Murat Özdoğan

September 2008

Thesis Advisor:	Arnold Buss
Second Reader:	Enver Yücesan

Approved for public release; distribution is unlimited
This thesis was done at the MOVES Institute

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, collecting and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2008	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Mission Assignment Model and Simulation Tool for Different Types of Unmanned Aerial Vehicles			**5. FUNDING NUMBERS**
6. AUTHOR(S) Yücel Alver Murat Özdoğan			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The use of unmanned aerial vehicles on the battlefield becomes more and more important every day. Parallel to this growing demand, there is a need for robust algorithms to solve the mission assignment problem in an optimum way. There are several tools for solving the assignment problem and testing the results to evaluate the robustness of the proposed algorithm. For most of the models, input factors are limited to the most important ones to make the process simpler. The aim of this thesis is to create an optimal solution for the assignment problem and test its robustness with a stochastic simulation tool. To accomplish the goals more factors, such as ground abort rates of the UAVs and the area weather risk levels are added. These factors, which were typically excluded from previous studies, are incorporated to make the model more realistic. The analysis and the results proved that the assignment algorithm works well and creates plausible results.

14. SUBJECT TERMS UAV, Model, Simulation, Assignment Problem, Simkit, Regression Analysis.			**15. NUMBER OF PAGES** 100
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

MISSION ASSIGNMENT MODEL AND SIMULATION TOOL FOR DIFFERENT TYPES OF UNMANNED AERIAL VEHICLES

Yücel Alver
Captain, Turkish Air Force
B.S., Turkish Air Force Academy, 2000

Murat Özdoğan
1st Lieutenant, Turkish Air Force
B.S., Turkish Air Force Academy, 2000

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN
MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION (MOVES)

from the

NAVAL POSTGRADUATE SCHOOL
September 2008

Authors: Yücel Alver

Murat Özdoğan

Approved by: Arnold Buss, Ph.D.
Thesis Advisor

Enver Yücesan, Ph.D.
Second Reader

Mathias Kölsch, Ph.D.
Chair, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The use of unmanned aerial vehicles on the battlefield becomes more and more important every day. Parallel to this growing demand, there is a need for robust algorithms to solve the mission assignment problem in an optimum way. There are several tools for solving the assignment problem and testing the results to evaluate the robustness of the proposed algorithm. For most of the models, input factors are limited to the most important ones to make the process simpler. The aim of this thesis is to create an optimal solution for the assignment problem and test its robustness with a stochastic simulation tool. To accomplish the goals more factors, such as ground abort rates of the UAVs and the area weather risk levels, are added. These factors, which were typically excluded from previous studies, are incorporated to make the model more realistic. The analysis and the results proved that the assignment algorithm works well and creates plausible results.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF ABBREVIATIONS AND ACRONYMS

AAA	Anti Aircraft Artillery
ATO	Air Tasking Order
BDA	Battle Damage Assessment
DES	Discrete Event Simulation
ESM	Electronic Support Measures
FEL	Future Event List
GCS	Ground Control Station
GUI	Graphical User Interface
METAR	Meteorological Terminal Aviation Routine Weather Report
NBC	Nuclear, Biological and Chemical
NOLH	Nearly Orthogonal Latin Hypercube
RTB	Return to Base
SAM	Surface to Air Missile
TAF	Terminal Area Forecast
TOW	Target Opportunity Window
TSP	Traveling Salesman Problem
UAV	Unmanned Aerial Vehicle

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The first Unmanned Aerial Vehicles (UAVs) were deployed in the 1950s. During the early years of operation, they were very few in number and mostly used for surveillance missions. With the rapid evolution of the technology and the fast growth in demand, UAVs are being used for various missions in nearly every operation. As air tasking orders are created prior to the operation, in order to prevent conflicts between aircraft, there becomes a need for UAVs to be assigned accordingly. Since the number of UAVs to be used in an operation is very high, the assignment problem has to be solved with computer algorithms. There are several techniques and algorithms that seek to deal with this problem. However, there are so many different inputs and constraints to take into consideration, that it is impossible to identify one optimum solution for the problem.

To overcome the problem of assigning UAVs to the missions, there are several approaches. Branch and bound algorithm is one of them and seems to be the best since it searches for all possible assignment combinations. This seems like an appropriate approach in order to find the optimum solution, but the process requires a huge computing time. As the number of UAVs increase, so too will the possible assignment combinations. Therefore, the optimum solution has to address both operational and computational needs.

This thesis seeks to find an optimum solution for the mission assignment problem of different types of UAVs. As mentioned above there is no single optimum solution; therefore, the algorithm created in this thesis tries to solve the problem in an optimum manner within a plausible computing time. Another purpose of this thesis is to add as many input factors as possible, to create a more realistic and robust model. Most of the models capture only the most important factors to keep the model simpler and faster. On the other hand, some of the factors are considered to be less important and are not included in most of the earlier models. After solving the assignment problem in an optimal way, the

execution phase will evaluate the robustness of the assignment algorithm. To accomplish this, the authors simulated a full cycle in a UAV operation. This consisted of starting from the assignment phase, going through all the steps in preflight activities, traveling to the mission area, conducting mission and post-mission activities and, finally, returning the UAV to base and finishing the maintenance to make it ready for the next missions.

For running the simulation, 15 input factors and eight performance measures were created. Nearly Orthogonal Latin Hypercube (NOLH) design was used to determine the design points, resulting in 129 design points capturing the variations of 15 input factors. The simulation was run for 100 replications to produce enough outputs for making reasonable output analysis.

The output analysis was conducted in two sections. For both sections, regression analysis is used as a primary tool. When needed, partition trees are used to analyze the effects of some of the factors. In the first section, the effects of the main factors were analyzed to understand which factor or factors had the most impact on the performance measures. As mentioned earlier some of the factors that seemed to be unimportant turned out to be the most or second-most important factors on some of the performance measures. In the second section some of the key interactions beyond the main factors were analyzed. Key interactions are crucial since some of the factors become important only when they are in interaction with some other factors. Therefore, the analyst can capture the effects of some factors only by observing the interaction plots.

Finally, the analysis and the results proved that the assignment algorithm works well and creates plausible results. Since all the data used were generic, the authors cannot claim that the results will provide insights about real life situations. However, as stated before, the goal of this thesis is to create a template model that can be modified with real life data. Therefore, by replacing the input factors with real data, the decision makers can use this model to help them make decisions about UAV assignment problems. The model will serve for a variety of other areas, such as deciding the UAV demand if the possible

missions are known. In addition, the maintenance issues will be observed and possible solutions will be created by analyzing the maintenance queue wait times.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

We would like to thank Professor Arnold Buss and Professor Enver Yücesan for their help and guidance throughout the preparation of this thesis. We also would like to thank Professor Susan Sanchez for her valuable support. Their valuable inputs and instructions helped us stay focused.

Yücel Alver:

Finally and most importantly, I would like to express my gratitude to my lovely wife, Burcu, for her support, patience and understanding throughout all phases to complete this thesis. She was there whenever I needed and without her, it would not be possible to finish this thesis. In addition, I would like to thank my parents and my brother for their support from overseas.

Murat Özdoğan:

I also would like to express my appreciation and gratitude to my wife, Dilek, for her patience, understanding, and support throughout my study. Her presence made it possible to finish this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. GENERAL

The rapid pace of technological change has caused the systems used on the battlefield to evolve continuously. As part of this evolution, scientists are searching for ways to make the systems work autonomously, in other words, to decrease the dependency on human operators. Unmanned Aerial Vehicles (UAVs) are now taking the place of manned vehicles, especially for intelligence purposes. Naturally, there are both advantages and disadvantages to this relatively new technology; but, above all else, UAVs eliminate or sharply reduce risk to the lives of their pilots or operators. Therefore, it can be said that the need for UAVs will increase at a rapid pace in the near future. Nearly all future combat systems require a huge number of UAVs involved in every stage of the mission, from pre-combat intelligence gathering to post-combat Battle Damage Assessment (BDA).

In the early days of UAV usage, there was no need for complex assignment or scheduling algorithms for mission assignment problems, since there were only a limited number of UAVs. Today there are many different types of UAVs performing a variety of missions. Therefore, to ensure mission effectiveness, well-designed simulation models are necessary, first to determine the needs and then to effectively assign and deploy the UAVs.

B. PURPOSE

The main goal of this thesis is to create an efficient algorithm for solving the assignment problem of different types of UAVs—with different attributes and constraints affecting possible assignments—to missions, and to test the solution's robustness through designed simulated experiments. In the first (planning) phase, the idea is to create an Air Tasking Order (ATO) kind of list for the next day's missions with the available resources. While solving the assignment problem, the purpose is to find a near optimal solution that deploys

the resources most effectively. After solving the assignment problem, the second phase is to create a simulation package to include the execution phase of these assigned missions. In this phase, the simulation has to be as realistic as possible to test the robustness of the assignment algorithm in the presence of both deterministic and stochastic elements, in order to capture the known facts and the uncertainty of real life operations.

According to the goals stated above, the first step is to identify relevant performance measures and then conduct a detailed output analysis to determine the effects of the input factors on these performance measures.

C. BACKGROUND

Since the focus of this thesis is UAV systems, this chapter will present some background knowledge about UAVs and their support systems. To start with, it is necessary to understand how and why UAV systems increasingly are being used. Therefore, this chapter lists the advantages of this relatively new technology compared with manned vehicles.

UAVs were first used in the 1950s. At first, two main advantages were considered over manned aircraft. The first and most important advantage is that, since there is no pilot in these vehicles, there is no direct risk to human life with their use. The second and also very important point is that these vehicles are cost effective when compared with manned aerial vehicles. As the technology evolved, another important advantage emerged: since there is no human on board, the vehicle is not bound by human limitations (Geer & Bolkcon, 2005). For example, most fighter aircraft are limited to a certain *g* level because of their pilots' *g* limits. In addition, it is unimaginable for a fighter aircraft to fly 50 hours continuously. However, *g* limits or endurance is not an important issue for a UAV.

After being referred to under such different names as “remotely piloted vehicles” or “pilotless aircrafts,” today these vehicles are referred as “unmanned aerial vehicles” (Geer & Bolkcon, 2005). UAVs can be defined as remotely piloted or autonomous aerial vehicles that can carry sensors, cameras, and a

variety of payloads; these payloads are mostly used for intelligence gathering, reconnaissance, target acquisition, and battle damage assessment missions (Pike, 2007).

UAVs are generally classified in three main categories according to their endurance, altitude, and role. Their endurance is classified as short, medium, and long. Their altitude is classified as low, medium, and high. Finally, alphanumeric codes are used to specify their role: C is used for cargo, R is used for reconnaissance, M is used for multi-role, and Q is the general designation for unmanned aerial systems (Headquarters Department of the Army, 2006). For example, the U.S. Air Force's medium altitude, long endurance Predator UAV is referred to as MQ-1 showing that it is a multi-role unmanned aerial vehicle.

For this thesis, the Gnat and Heron types of UAVs are considered in the simulation model. They are both medium-altitude, long-endurance and multi-role UAVs. In the following sections, further information about these two types and their supporting systems are provided.

1. Gnat

General Atomics' Gnat is a medium altitude, long endurance UAV (Pike, 1999). The first model, Gnat 750, is derived from the earlier Amber program and has been flying since 1989 (*Gnat 750.*). In addition to its long endurance, it can also carry large payloads. Gnat 750 is the pioneer in UAV technology in many ways. It was the first UAV controlled via satellite in 1992; with its endurance over 40 hours, it can therefore be operated at very long ranges (*GA-ASI Gnat*, 2007). I-Gnat is a newer version of Gnat 750 with additional capabilities to improve its performance. Below is a picture of I-Gnat in flight.

Figure 1. I-Gnat

There are four types of Gnats in service. Gnat A is the first UAV of this type and is called Gnat 750. In this thesis, Gnat D, also called I-Gnat, is modeled. Its specifications are as follows (*GA-ASI Gnat*, 2007):

- Power Plant: One 78.3 kW Rotax 914F turbocharged propeller
- Wing Span: 16.76 m (55 ft)
- Overall Length: 8.13 m (26 ft 8 in)
- Empty Weight: 513 kg (1,130 lb)
- Payload Capacity: 204 kg (450 lb)
- Maximum Take Off Weight: 1,043 kg (2,300 lb)
- Maximum Speed: 222 km/h (138 mph)
- Long Range Cruising Speed: 135 km/h (84 mph)
- Ceiling: 7,620 m (25,000 ft)

- Maximum Operational Radius: 2,778 km (1,726 miles)
- Maximum Endurance: >40 h

Other than its physical specifications, mission payloads have a very important role on dictating the capacities of the UAVs. Gnat can carry payloads for surveillance, reconnaissance, Electronic Support Measures (ESM), Nuclear Biological and Chemical (NBC) detection, and radio relays (*GA-ASI Gnat*. 2007). Different types of payloads and the problems related to assigning or operating these payloads are not included in the model created for this thesis.

2. Heron

Like the Gnat, IAI/Malat's Heron is also a medium altitude, long endurance UAV. It started service in 1994 to replace the IAI Searcher Mk I and Mk II models. In addition to its long endurance of up to 52 hours, one of the most important improvements of this UAV is its fully autonomous feature. It has automatic take-off and landing capability, and all the missions can be pre-programmed for fully autonomous sorties (David, 2005). In other words, other than collecting and analyzing the data, there is no need to manually operate this UAV.

The Heron can carry a variety of payloads that make the Heron suitable for surveillance, reconnaissance, and many other missions, day or night. Its radar is capable of tracking 32 targets at a time, which makes it a very powerful system for surveillance missions (David, 2005). Figure 2 is a picture of the Heron in flight.

Figure 2. Heron

There are two types of this UAV, referred to as Heron-I and Heron-II. The specifications of Heron-I are as follows (*IAI Heron 1*, 2007):

- Power Plant: One 73.5 kW turbocharged Rotax 914 F propeller.
- Wing Span: 16.60 m (54 ft 5.5 in)
- Overall Length: 8.50 m (27 ft 10.6 in)
- Payload Capacity: 250 kg (551 lb)
- Maximum Take Off Weight: 1,100 kg (2,425 lb)
- Maximum Speed: 231 km/h (144 mph)
- Long Range Cruising Speed: 130 km/h (81 mph)
- Ceiling: 8,075 m (26,500 ft)
- Maximum Range: 1,000 km (621 miles)
- Maximum Endurance: > 40 h

3. Ground Control Station

Ground Control Station (GCS) is one of the most important components of the UAV systems. Since the UAVs are controlled by an operator and the collected data has to be analyzed, the GCS can be referred to as the backbone of the whole system. There are three main functions of GCS: mission planning, mission control, and data collection/manipulation (Anderson, 2002). There has to be at least one operator to conduct a UAV mission. Generally, there are one or more personnel in charge of each GCS function. Figure 3 is a general view of a GCS.

Figure 3. Inside view from a stationary GCS

Mission planning is a critical function for identifying the correct UAV with correct payloads to accomplish the desired missions. Usually, based upon the surveillance collected from other units, mission planners select the type of the UAV and the sensors to be used. The area threat level is taken into account in this phase in order to avoid vehicle loss. Mission prioritizing is another key decision in this phase. There is usually a restriction on the number of UAVs to be operated by one GCS at a time, which is dictated by the communication capabilities of the GCS and the type of the UAV. The newer systems allow more UAVs to be operated at a time, but there is still a limit. Once the planning is completed, the UAV operators take charge.

Mission control starts with powering on the UAV and ends with recovering it after landing. Depending on the mission duration, there will be one or more operators at the GCS. Although there are UAV systems that are fully autonomous with take-off, mission execution, and landing capabilities, the operator is always in charge of making the corrections or changes that might occur any time. In some systems, the operator flies the UAV with the camera attached to it, while other systems use consoles that simulate the route of the UAV on a digital map in real time. The operator is also in charge of the payloads attached to the UAV. The ground control stations can be stationary or mobile (to be moved anywhere as needed). Figure 4 shows a mobile GCS mounted on a truck. These systems can also be transported on cargo planes such as the C-130.

Figure 4. A Mobile GCS

The last function is data collection and manipulation. To achieve the goal of the UAV, the collected data has to be transferred to the GCS simultaneously. After collecting the data, the personnel have to exploit this information according to the mission requirements. After that, the data must be stored for further use or archiving purposes.

D. OUTLINE OF THE THESIS

The thesis consists of six chapters. The first chapter covers the main idea behind this thesis, the purpose of the study, and the background about UAVs and their supportive systems. In the second chapter, modeling tools, input factors, performance measures and assumptions are described in detail. The third chapter is about model description, and includes the information about the assignment problem and the execution phase. The fourth chapter covers the details on design of the experiments. The factors, design points and replication issues are explained in this chapter. The fifth chapter is the results and output analysis chapter, which discusses the effects of the input factors on the performance measures and comes up with some possible optimum solutions for such problems. The last chapter summarizes the results and points out possible future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MODELING TOOLS

This chapter describes Discrete Event Simulation (DES), Simkit and Nearly Orthogonal Latin Hypercube (NOLH) as the modeling tools used to create the simulation model for this thesis.

A. DISCRETE EVENT SIMULATION

Simulation tools can be classified in two categories, according to their time advance mechanisms: discrete event simulations and continuous simulations. In Discrete Event Simulation (DES), the state of a system changes instantly at distinct time points (Schriber & Brunner, 2005). In other words, the system clock is advanced only to those discrete points in simulated time where the next event, which may change the system state, is scheduled to occur. In continuous simulations, system state evolves continuously, as the simulation clock is advanced in small, fixed time steps. Since the value of simulated time is important in DES, an internal variable, called as simulation clock, keeps track of time and advances in discrete steps. At the beginning of the simulation, time is initialized to zero. The simulation clock then advances to the next event time and updates the stated variables. Then the simulation clock jumps to the first event in the event list. Every DES has an event list that contains a set of events that are ordered in time sequence. This event list is called the Future Event List (FEL). System state does not have to be changed between the time advances. Events have to be in time order; otherwise, events would be triggered out of order and cause the time to go backward, which is not acceptable in real life. The time advances continue until a predefined stopping condition is met or the last event in FEL has been executed (Law, 2007). The simulation clock and the real clock (the clock on the wall) are completely different. The real clock advances second by second continuously, while the simulation clock jumps from one event time to another. Moreover, these jumps do not have to be equal in size.

B. SIMKIT

Simkit is a software package that enables implementation of DES models. Simkit is developed and maintained by Professor Arnold Buss. This Java-based package "is oriented towards Event Graph Methodology" (A. Buss, 2005). Event Graph Methodology (Schruben, 1983) is powerful for its expressiveness, simplicity, and extensibility for constructing and representing DES programs.

There are four basic elements for event graphs: parameters, state variables, events, and scheduling edges. State variables are the elements that have the possibility of changing through the simulation run. State variables define the state of the system. Parameters are the elements that are constant and never change in the course of simulation. Random variables are considered to be parameters. An event can be defined as "an instantaneous occurrence that may change the state of the system" (Law, 2007). Events are the labels of state transition functions. The collection of all the events in a simulation provides all the possible value changes. Scheduling edges define the logical and temporal relationships between events.

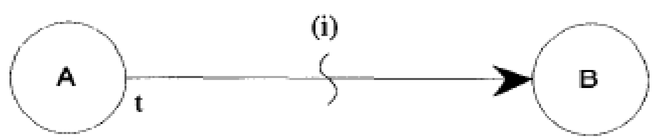


Figure 5. Basic Event Graph Model.

Figure 5 (Schruben, 1983) represents a scheduling edge between event A and event B. This scheduling edge is interpreted as follows: event B is scheduled to occur *t* time units in the future after event A occurs and if the condition (i) is true.

The following Java code is the implementation of Figure 5 in Simkit:

```
public void doA() {

        <code to perform state transition for event A>
```

```
		if (i) {
			waitDelay("B", t);
		}
	} (A. Buss, 2001)
```

In Event Graph Methodology, “every event graph has at least one Run event, and that event is assumed to be placed in the Event List at time 0.0” (A. Buss, 2005). If the model does not recognize a Run event at the beginning, simulation stops immediately. Since the event list is empty at the beginning of the simulation, the Run event provides a starting point for the model. The Run event also initializes the state variables and schedules the next event (Buss, 1996). Table 1 represents the correspondences between elements in an event graph and Simkit (A. Buss, 2005).

Table 1. Event Graph/Simkit Correspondence

Event Graph	**Simkit**
Parameter	Private instance variable
State Variable	Protected instance variable
Event	“do” Method
Scheduling Edge	Call to “waitDelay”

In Simkit, parameters are implemented as private instance variables while state variables are implemented as protected instance variables. Every method that corresponds to an event starts with a string “do”. Scheduling edges are implemented by a call to a waitDelay() method. When waitDelay() method is called by the program, an event is created and added to FEL. This method takes at least two arguments: the first argument represents the event name and the second argument represents the time delay.

C. NEARLY ORTHOGONAL LATIN HYPERCUBE (NOLH)

Valid modeling must be coupled with efficient experimental design for effective simulation analysis. There are many experimental design methods in literature. The most commonly used design is the factorial design. A 2^k factorial design requires only two levels (low and high) for each factor and is easy to construct. Factorial designs are orthogonal and allow for examination of more than one factor at a time. Researchers can determine the main effects of several factors and the interactions between them. Despite all the advantages of using factorial design, it may not be good enough for some experiments. Without replication, for instance, a 2^k factorial design may not be good enough for estimating all the effects, since no degrees of freedom remain for error. Moreover, if there are a large number of factors in an experiment, the required data grows dramatically. Table 2 represents the required number of design points for different numbers of factors.

Table 2. Number of Design Points Required for 2^k Factorial Design

Number of Factors	**2^k factorial**
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$
4	$2^4 = 16$
5	$2^5 = 32$
10	$2^{10} = 1,024$
15	$2^{15} = 32,768$
20	$2^{20} = 1,048,576$

Another drawback of 2^k factorial design is that it may not provide any information on how the simulation behaves in the interior points of the experimental region, since it runs the simulation at only two levels (low and high)

for each factor (Sanchez, 2006). An m^k factorial design can solve this problem by filling more space of the interior of the experimental region.

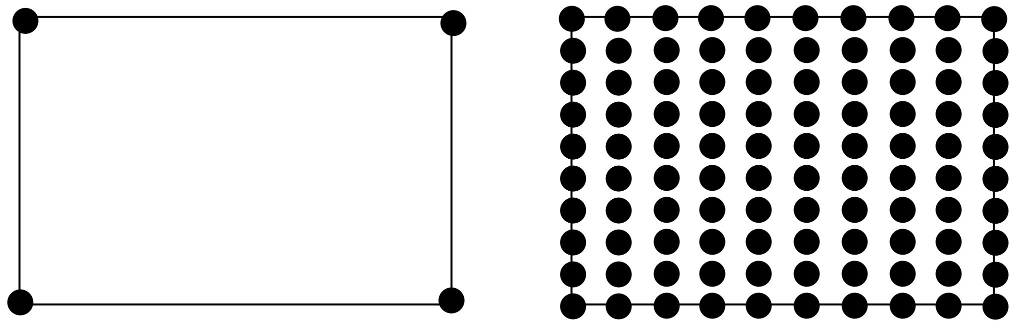


Figure 6. 2^2 and 10^2 Factorial Design

In an m^k design, *m* represents the number of levels of each factor and *k* represents the number of factors. In Figure 6, 10^2 means that there are two factors (X-axis of the plot represents the first factor and Y-axis represents the second factor) and each factor can have ten levels. A 10^2 factorial design reveals more information about the interior part of the experimental region. On the other hand, massive information requirement is still a big problem. Table 3 represents the required number of design points for different numbers of factors.

Table 3. Number of Design Points Required for m^k Factorial Design

Number of Factors	**10^k factorial**	**5^k factorial**	**2^k factorial**
1	10	5	2
2	100	25	4
3	1,000	125	8
4	10,000	625	16
5	100,000	3,125	32
10	10^{10}	9,765,625	1,024
15	10^{15}	> 30 billion	32,768
20	10^{20}	> 95 trillion	1,048,576

The Nearly Orthogonal Latin Hypercube (NOLH), a space filling design, provides more efficiency and flexibility compared to a factorial design. Cioppa and Lucas (2007) developed NOLH design tables that have good orthogonality properties and provide some information about the behavior of the model at the interior points of the experimental region. In addition, the data requirement for NOLH designs is less than in the factorial design. Table 4 shows the number of factors and the associated number of design points.

Table 4. Number of Design Points Required for NOLH Design

Number of Factors	Number of Design Points
2 - 7	17
8 – 11	33
12 – 16	65
17 – 22	129
23 - 29	257

III. MODEL DESCRIPTION

The main goal of this thesis is to create a tool that models the planning and execution of a one-day scenario for several UAVs to accomplish a number of missions with different attributes. Therefore, the model consists of two phases. The first phase is the assignment and the second is the execution. The assignment phase starts at simulation time zero to collect all the candidate missions for the next day. The missions are created at simulation time zero, to be executed starting from 0700 of the next day and within a 24-hour period. Once all the missions are created and saved into the mission list, the assignment algorithm is run to assign these missions to the UAVs in an optimal fashion. The assignment algorithm is also run at simulation time zero. Therefore, the ATO for UAVs is created one day prior to the execution phase at simulation time zero. Once the assignment phase ends, the execution phase starts with the preflight inspection of the UAVs. In the ATO, the UAVs are ordered in a mission list according to their first mission's preflight inspection time. The UAVs could have more than one mission assigned to them. Therefore, the execution phase is triggered by the first UAV's first mission's preflight inspection time. The other assigned UAVs follow the first one when their preflight inspection time is reached. The execution phase ends when the last UAV returns to the base and its maintenance is completed. Since the missions can start within a 24-hour period, the execution phase may take more than 24 hours, including the time for the last UAV to return to base and the time for its maintenance to be completed. Figure 7 shows the event graph of the entire model with its assignment and execution components.

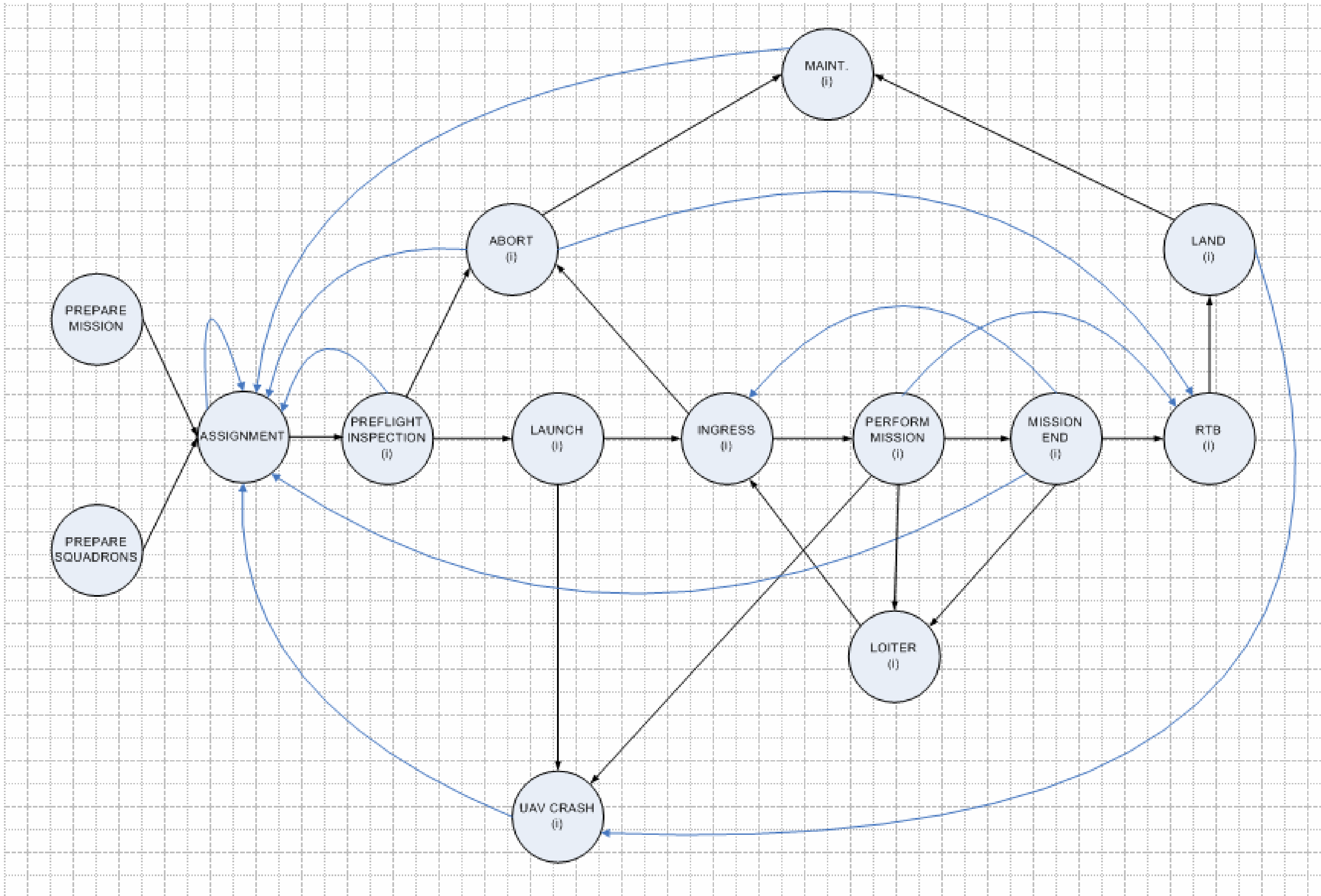


Figure 7. Event Graph of the Model

A. ASSIGNMENT PROBLEM

Before starting to explain the method used for solving the assignment problem, it is necessary to mention some of the characteristics of the model that was created. First, a UAV can be assigned to more than one mission. This is the key factor for the assignment problem; if a UAV could only be assigned to one mission, the problem would be much easier. On the other hand, a UAV can only be assigned to a limited number of missions since there are constraints for both the missions and the UAVs. The most important constraint about the missions is that they have a target opportunity window. The missions must be started and finished within that time window. For the UAVs, the most important constraint is their endurance. The UAVs will fly only for a specific amount of time because of the fuel constraint. Therefore, the model has to assign missions to the UAVs in an optimum way. There will be different approaches to solving this problem. For example, maximizing the assigned missions' number will be one solution. Nevertheless, in real operations it is more important to use the limited resources to complete the critical missions than it is to complete more but unnecessary missions. Therefore, while creating the missions, the authors assign them a bonus point to express their importance. After all, the purpose can be specified as assigning missions to the UAVs with maximum possible bonus points.

After defining the problem, the authors searched for the methods to solve it. This problem is similar to the very well known Traveling Salesman Problem (TSP). The TSP algorithms try to find the best traveling route for visiting the cities and returning to the original destination. In the model for this thesis, the missions are in locations other than cities, and there are UAVs instead of traveling salesmen.

One of the most common tools for solving the TSP is the branch and bound method (Radharamanan & Choi, 1986). The branch and bound algorithm solves discrete optimization problems by searching all the possible combinations. However, the implementation of the branch and bound method to the UAV problem raised other issues. In Figure 8, the search space for three missions can

be seen. There are three factorial combinations to be compared, which will not take much time; however, if there are 30 or 40 missions in the mission list then the number of mission combinations jumps to very high numbers. For example, for 40 missions there are 40 factorial combinations to be compared. Even with the very fast computers available today, it will take serious computing time to solve this problem. Since the assignment module is intended to create the ATO for the next day's missions, it is unbearable to run this module for more than two or three hours. Otherwise, it will not be realistic to wait for hours since there are so many things to do before performing the missions.

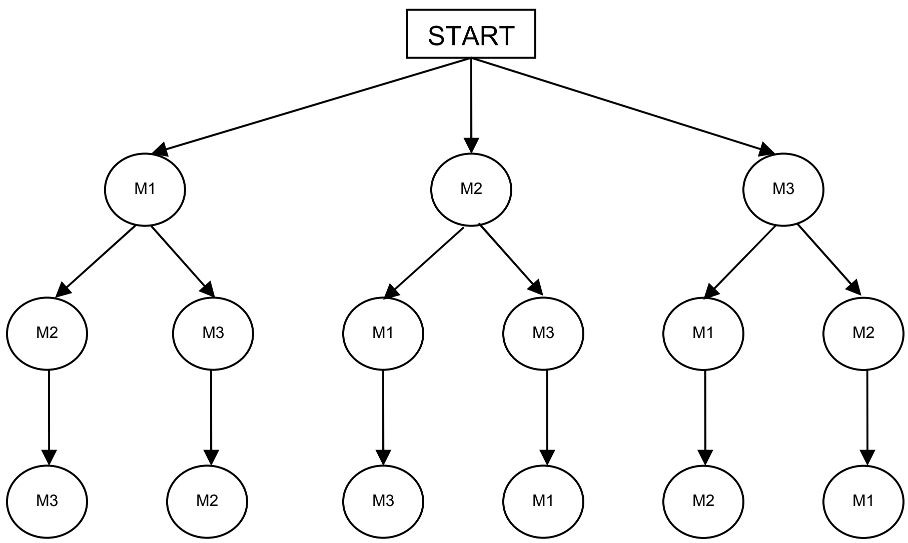


Figure 8. Search Space of Branch and Bound.

To overcome the computing time problem, the authors developed a different algorithm for getting similar results to the branch and bound algorithm, but within a plausible computing time. For the algorithm used in this thesis, the goal is not to compare all the combinations. Instead, the goal is to find out the optimum combination "on the fly." The authors start with the first mission in the mission list, then check a series of statements to find out if the UAV can be assigned to that mission or not. The statements are:

1. Maximum operational range of the UAV has to be longer than the distance between the mission location and the base.

2. The UAV has to perform the mission within the given time period. It has to be in the mission area after TOW start time and has to accomplish the mission before TOW end time.

3. The UAV has to be able to return to base after executing the mission. UAVs have limited fuel for endurance. The UAV is considered to have a full tank of fuel at the beginning of the simulation. Fuel amount decreases as the UAV stays in the air.

4. The UAV has to have reserve fuel that will be enough for 60 minutes flying after returning to the base. This reserve fuel is considered to be the emergency fuel that can be used in case of situations where the runway is closed for any reason and the UAV has to divert to an alternate airfield.

If the UAV meets all these requirements, the first mission in the assignment mission list is added to the mission list of that UAV, the location of the UAV is changed to the mission location, the remaining fuel of the UAV is decreased accordingly and the current time is recalculated.

Based on the current time, and the location and remaining endurance of the UAV, the algorithm checks if the UAV can accomplish the second mission in the assignment mission list. If the UAV meets the requirements, explained above, for the second mission, that mission is added to the mission list of the UAV. However, the important thing at this point is that even if the UAV can accomplish this second mission after the first one, the location and the endurance of UAV are not changed. Since the algorithm is designed to find the next mission that can be accomplished and has the highest bonus point, the model keeps searching for the alternate missions that can be accomplished with higher bonus points. For example, if the UAV can be assigned to the second, fifth and ninth missions after accomplishing the first mission, the mission with the highest bonus among them

stays in the mission list of the UAV. After finding the next optimum mission, the current time, location and remaining endurance of the UAV is recalculated.

This process continues in the same manner through all the missions in the assignment mission list. At the end of this process, a reasonable route that starts with the first mission is found. The total bonus that the UAV can earn by accomplishing these missions is calculated by adding the bonus points of each mission in the UAVs mission list. Then the program resets everything and the same process starts from the beginning, but from the second mission in the assignment mission list instead of the first one. At the end of this process, another mission list is created and total bonus points are calculated. The process continues by starting from each mission. At the end, the algorithm compares these total bonus points and assigns the route that has the highest bonus to the UAV. Figure 9 shows an example of how the assignment process works. In this example there are two possible routes: one starting from M1 and the other from M9. The total bonus earned for the first route is 35 plus 75 plus 90 equals 200, and for the second route is 60 plus 80 plus 55 equals 195. Therefore, the algorithm selects the first route and assigns M1, M5 and M6 to the UAV.

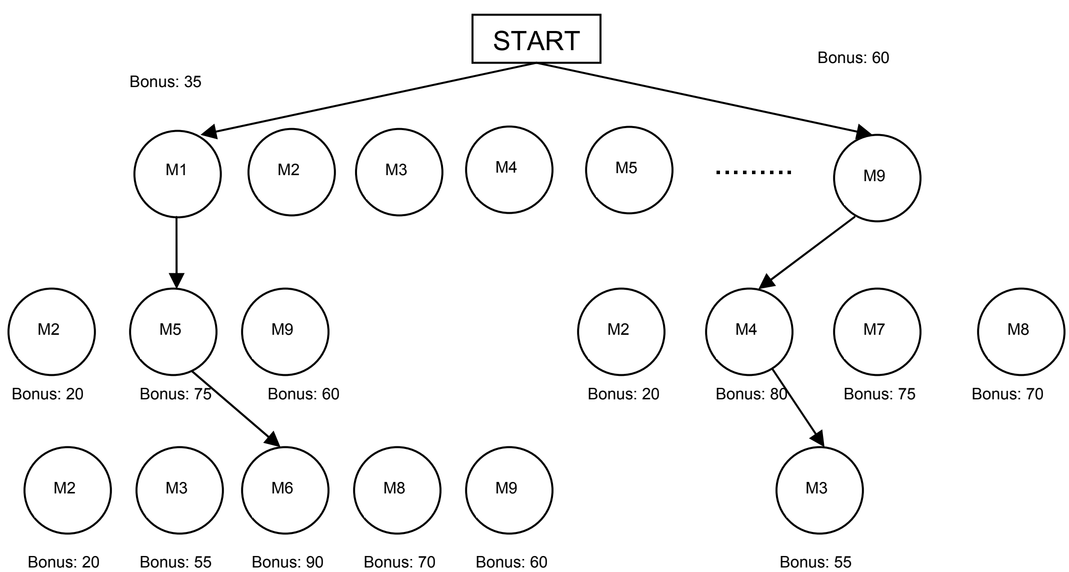


Figure 9. Search Algorithm for Assignment Problem

Since there are two different types of UAVs, the same process is completed for the other type of UAV. Therefore, at the end of a full run of the assignment phase, there will be two lists, one containing the missions with the highest points for Heron and one containing the missions with the highest points for Gnat. Finally, the algorithm selects the UAV and route combination that has the highest bonus points. The assignment process continues until either all the missions in the assignment mission list are assigned or there are no more UAVs to be assigned.

When compared to the results of the algorithm used with the branch and bound algorithm, the results seem to be the same for most cases. The branch and bound algorithm works better in a very few situations, since it is designed to search for all possible combinations. However, while creating a simulation tool it is very important to make it robust, but usable. For more missions in the mission list, the branch and bound algorithm will take days to complete the search. Even though it creates results that are more robust, these data will not help the decision makers plan the next day's missions.

B. ASSUMPTIONS

1. Both types of UAVs are capable of accomplishing all types of missions in the mission list.

2. Both types of UAVs take off with sufficient pods to accomplish the missions assigned to them for that sortie.

3. There are no crew related limitations. There are a sufficient number of operators and mechanics to keep the two bases in 24-hour operation.

4. The meteorology station's accuracy for predicting the mission area weather is 95 percent.

5. There will be no communication disruption between the ground control stations and the UAVs within their ranges.

6. There is a sufficient number of ground control stations to operate all the UAVs at the same time.

7. “Wear and tear” for UAVs is not included in the model. UAVs are “as good as new” after they leave the maintenance server.

8. UAVs can be assigned without delay to any mission after leaving the maintenance server.

9. Minimum landing fuel is considered to be that amount needed to fly a UAV for 60 minutes to the nearest airfield in case of emergency. If the main base of a UAV is closed for landing, the UAV is assumed to have enough fuel to fly to the nearest air base for a safe landing.

10. Preflight inspection time is 30 minutes for Gnat and 20 minutes for Heron.

11. Acceptable weather risk level is 0.8 for Heron and 0.85 for Gnat. Beyond these limits, the UAVs are assumed not to fly safely.

12. If a UAV aborts on the ground, it goes directly to maintenance. Time delays while transferring the UAV to the maintenance server are not considered in the model.

13. The operation starts 24 hours after running the simulation.

C. EXECUTION

As mentioned above, the execution phase starts with the first UAV’s first mission’s preflight inspection time. When the preflight inspection event is triggered for the first time, the simulation time is advanced to the preflight inspection time of the first UAV and the discrete event simulation starts. After this point, all the events are called according to the times stated in the waitDelay methods of prior events. A UAV must complete at least nine events to complete a mission cycle. However, this number can change under specific situations such as a crash or an abort. Under normal conditions, after completing the preflight inspection, the UAV is launched and ingresses to the first mission area. Then it

completes the mission and ingresses to the other missions (if it has more assigned). After completing all the missions, the UAV returns to base, lands, and joins the maintenance queue. When its maintenance is completed, the UAV rejoins its squadron to wait for its next missions.

The next section will describe in detail the 11 events in the execution phase.

1. Preflight Inspection

When a UAV arrives for preflight inspection, a random number is generated for each mission in the UAV's mission list. These Uniform (0,1) random numbers represent the TAF report's accuracy. Normally, TAF reports represent the weather conditions for an area over the next few hours. However, in this model TAF reports represent the weather risk level for a UAV over the next few hours. These reports are not always 100% accurate. A value of 1.0 implies that the weather risk reported by TAF will be exactly the same as the real weather risk. If the random number is 0.95, that means that the TAF report underestimates the weather risk in the mission area. Variability of TAF accuracy is represented by a uniform distribution.

When a mission is generated in the simulation, it is generated with the real weather risk for the mission period. The drawn random number is multiplied by this value to determine the estimate of the TAF report for the weather risk in the mission area.

The UAV aborts if the estimate of the TAF report for any mission in the UAV's mission list is greater than the acceptable risk level for the UAV. When a UAV aborts because of the weather risk, that UAV returns to the squadron. All the missions in the UAV's mission list except for the mission(s) that cannot be accomplished because of the weather conditions are added to the mission assignment list. Moreover, a new assignment event is scheduled for the missions in the mission assignment list.

If the UAV does not abort, preflight inspection starts. Before the preflight inspection, another Uniform (0,1) random number is generated to determine if the UAV is going to abort because of a malfunction during inspection. If this drawn random number is greater than or equal to the preflight abort rate of that type of UAV, it aborts. The UAV may abort any time between the start and the end time of the inspection. If the UAV aborts, a random number is generated to determine when it actually aborts. This random number is multiplied by the predefined preflight inspection time of that UAV to determine the abort time.

If a UAV aborts because of a malfunction, its abort type is set to “ground abort,” maintenance is scheduled for that UAV, and all the missions are added to the mission assignment list. The assignment process is scheduled again for the missions in the mission assignment list.

If the UAV does not abort because of weather risk or malfunction, operators launch it after the inspection. Figure 10 represents the event graph associated with the preflight inspection.

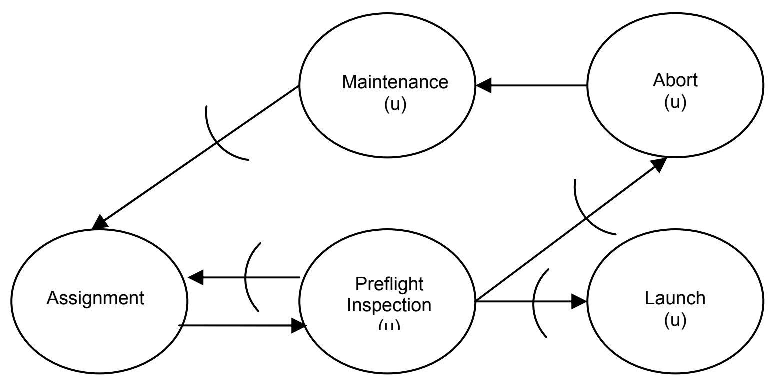


Figure 10. Event Graph for Preflight Inspection

2. Launch

Upon execution of the launch event, a random number is generated to determine whether the UAV is going to crash while launching because of operator failure or malfunction. This random number is compared with the

predefined crash rate of that UAV type. The UAV crashes at take off if the drawn random number is great than or equal to the crash rate of the UAV. When a UAV crashes, a “Uav Crash” event is scheduled in the simulation.

If the random number is less than the predefined crash rate of the UAV, an “Ingress” event is scheduled immediately, as seen in Figure 11.

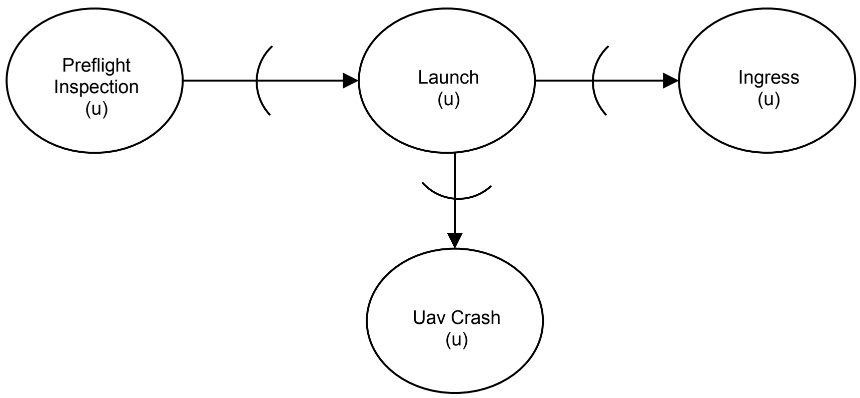


Figure 11. Event Graph for Launch

3. Ingress

An Ingress event is scheduled after a “launch,” “mission end” or “loiter” event. A random number is generated to define if the UAV is going to abort while traveling to the mission area. If the drawn random number is less than the air abort rate, the UAV arrives at the area for mission execution and a “perform mission” event is scheduled. Otherwise, the UAV aborts enroute and an “abort” event is scheduled. The UAV may abort at any time while traveling to the mission area.

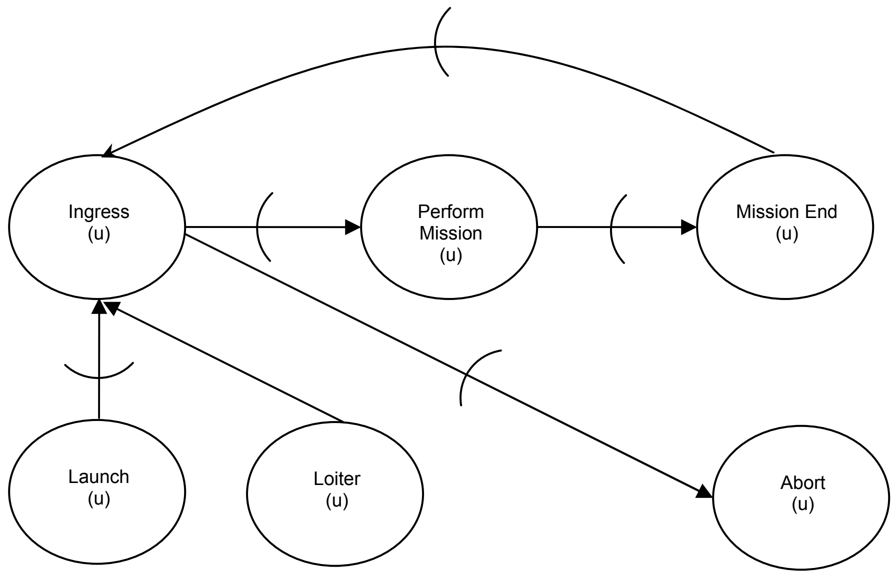


Figure 12. Event Graph for Ingress

4. Perform Mission

There are two risks for the UAV while performing a mission. These risks include attrition due to enemy fire and weather conditions. As mentioned before, UAVs are not scheduled for the missions that have high weather risk levels. However, since TAF is not able to predict the weather conditions with 100 percent accuracy, the UAV might be scheduled for a mission only to have the weather turn out to be too severe when the UAV arrives at the mission area. The real weather condition is stated while the mission is created. Therefore, when a UAV arrives at a mission area, the current weather conditions are checked. If the risk level is greater than the acceptable risk level, the UAV aborts that mission, which is added to the unaccomplished mission list. On the other hand, if there is at least one more mission in the UAV's mission list, a "loiter" event is scheduled for the UAV to spend the remaining time in a secure area. Otherwise, a "return to base" event is scheduled. If the weather risk is not an issue, then a random number is drawn to determine whether there would be attrition or not. If the drawn attrition probability is less than the threat risk level of the mission area, the UAV will be shot down while performing its mission. The attrition will occur any

time between the start time and the end time of the mission duration. If the UAV is shot down, a “Uav Crash” event is scheduled for the attrition time. Otherwise, the UAV completes that mission safely and an “end mission” event is scheduled.

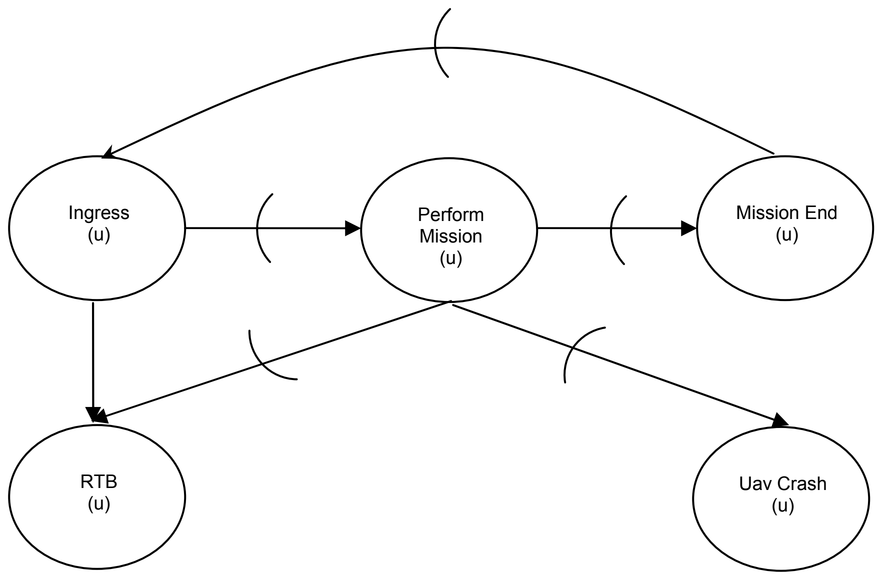


Figure 13. Event Graph for Perform Mission

5. End Mission

When a UAV accomplishes a mission successfully, an “end mission” event is scheduled. At the beginning of the “end mission” event, the accomplished mission is removed from the UAV’s mission list. If there is at least one remaining mission in the UAV’s mission list, the model checks whether this mission can be executed at the current time by that UAV, within the time constraints. Since preplanned mission durations can change in the air, this check has to be done before scheduling an “ingress” event. This check algorithm is described in the Assignment Problem section. The check algorithm not only decides if the UAV can be assigned to the next mission, but also returns a reason if the UAV cannot accomplish the next mission. The algorithm returns the following numbers depicting the reason:

0 : If the UAV arrives to the next mission area before the TOW start time for that mission.

1: If the UAV cannot accomplish the next mission before its TOW end time.

2: If the next mission's location is out of UAV's maximum operational range.

3 : If the UAV does not have enough fuel to perform the next mission.

If the UAV can perform the next mission, an "ingress" event is scheduled immediately. If the first mission in the UAV's remaining mission list cannot be executed by that UAV due to one of the above reasons, another event is scheduled.

Table 5. Scheduled Event According to the Reason

Reason	Scheduled Event
0	Loiter
1	Loiter / RTB and Assignment
2	N/A
3	RTB and Assignment

If the UAV arrives at the mission area before its TOW start time, it loiters in a safe area until the TOW start time. If the UAV does not have enough fuel to perform the next mission, a Return to Base (RTB) event is scheduled and the mission is added to the mission assignment list for a new assignment. If the UAV accomplishes the mission after the TOW end time, there are two options:

1. If there is at least one other mission in the UAV's mission list, the UAV loiters until the next mission's TOW start time in a safe area.

2. If there is no mission in the UAV's mission list, the UAV returns to base immediately.

In both situations the mission, which cannot be accomplished at the current time, is added to the mission assignment list and an "assignment" event is scheduled.

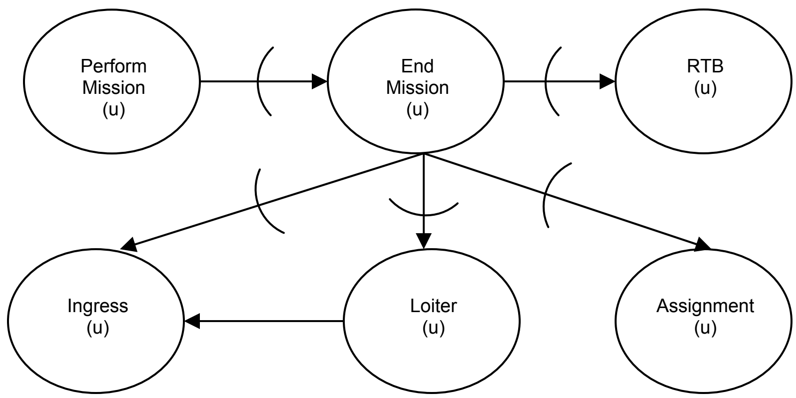


Figure 14. Event Graph for End Mission

6. Return to Base (RTB)

A Return to Base event is scheduled after "ingress" or "perform mission" events. When a UAV is ready to return to base, a random number is drawn to determine if the UAV is going to have a malfunction in the air. If the generated random number is less than the predefined air abort rate, the UAV returns to base safely and a "land" event is scheduled. Otherwise, the UAV's abort type is set to air abort, and after returning to the base and landing the UAV is scheduled for malfunction maintenance.

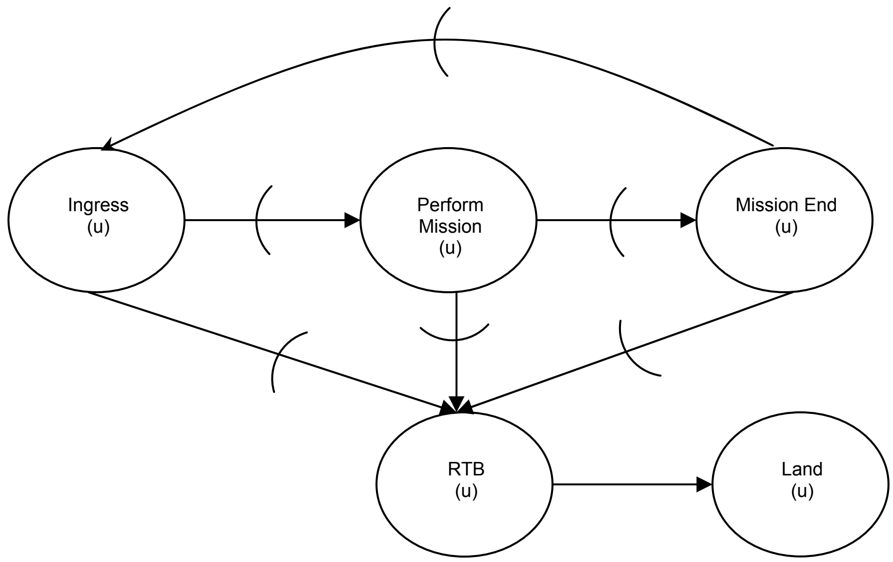


Figure 15. Event Graph for RTB

7. Land

A UAV lands after returning to base. When a “land” event is scheduled, a random number is generated to determine if the UAV is going to crash while landing. A UAV may crash because of a malfunction or operator failure. If the drawn random number is greater than or equal to the predefined land crash rate, the UAV crashes and a “Uav Crash” event is scheduled. If the UAV lands safely, a “start maintenance” event is scheduled.

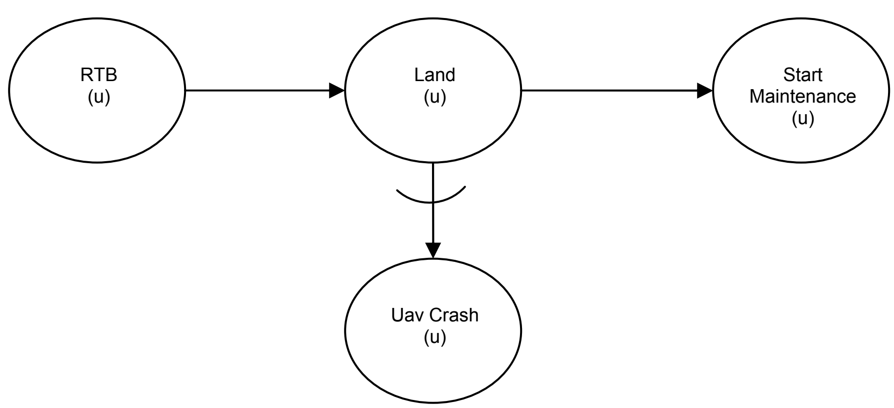


Figure 16. Event Graph for Land

8. Abort

When a UAV aborts, if there is at least one mission in the UAV's mission list, an "assignment" event is scheduled immediately. If it is a ground abort, the UAV goes directly to the maintenance server to get fixed. Otherwise, the UAV first has to return to base to get maintenance service. Whenever a UAV aborts, its maintenance service type is set to the malfunction maintenance.

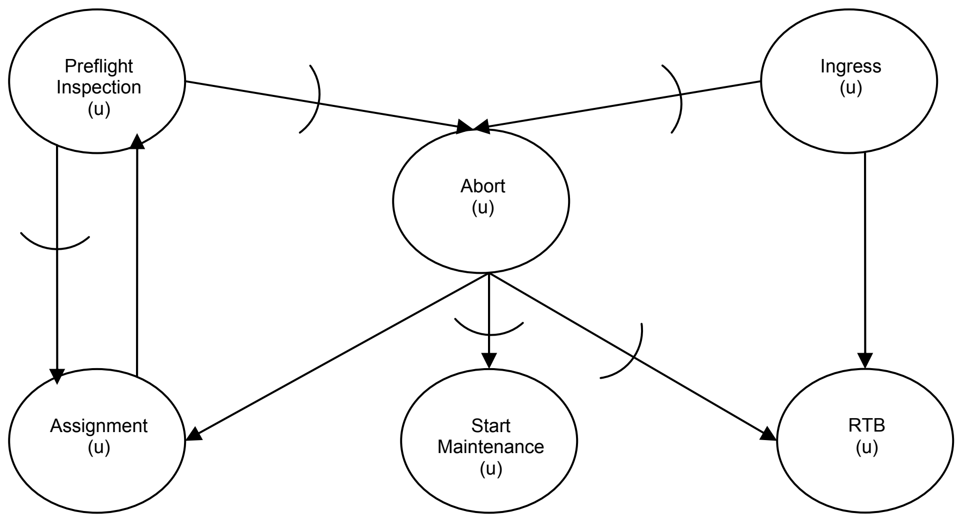


Figure 17. Event Graph for Abort

9. Loiter

This event is only scheduled after an "end mission" event. If the weather risk level of an area is higher than the UAV's acceptable risk level and the UAV has at least one more mission in its mission list, it loiters in a safe area to wait for the next mission. Loiter time of the UAV depends on the current time and the next mission's TOW start time. The loiter event schedules the "ingress" event when the time comes for the UAV to travel for the next mission's area.

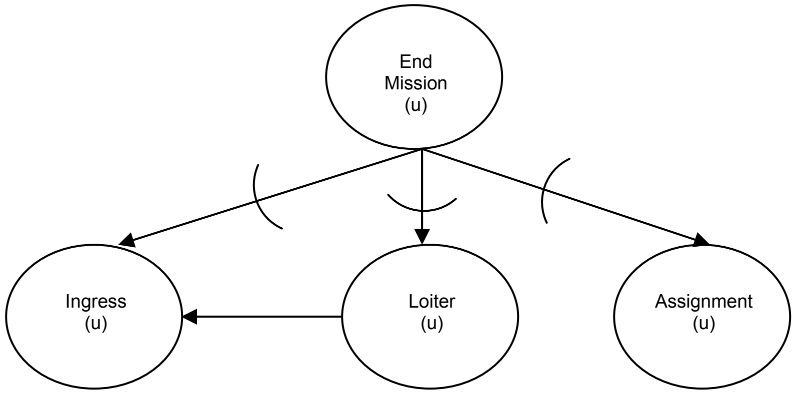


Figure 18. Event Graph for Loiter

10. Uav Crash

A UAV may crash while taking off, performing a mission or landing. A crashed UAV cannot return to its squadron and cannot be fixed. This event schedules an ”assignment” event immediately if there is at least one mission in the crashed UAV’s mission list.

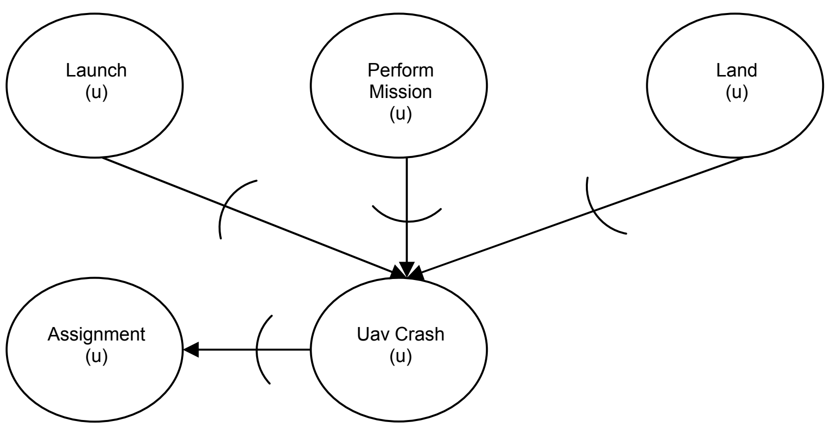


Figure 19. Event Graph for Uav Crash

11. Start Maintenance

A Start Maintenance event is scheduled after a “ground abort” or “land” event. There must be an available server to start the maintenance. If a UAV

aborts on the ground or in the air, it requires malfunction maintenance. If it lands at the base without any malfunction, it is scheduled for periodic maintenance.

Periodic maintenance durations of the UAVs are constant and set to 60 minutes for Heron and 80 minutes for Gnat. Malfunction maintenance durations of UAVs are random variables whose distributions are specified by the user before the simulation run. After a periodic or malfunction maintenance, UAVs are considered to be as good as new.

After the "start maintenance" event, an "end maintenance" event is scheduled immediately. If there is any remaining mission in the assignment mission list, an "assignment" event is scheduled.

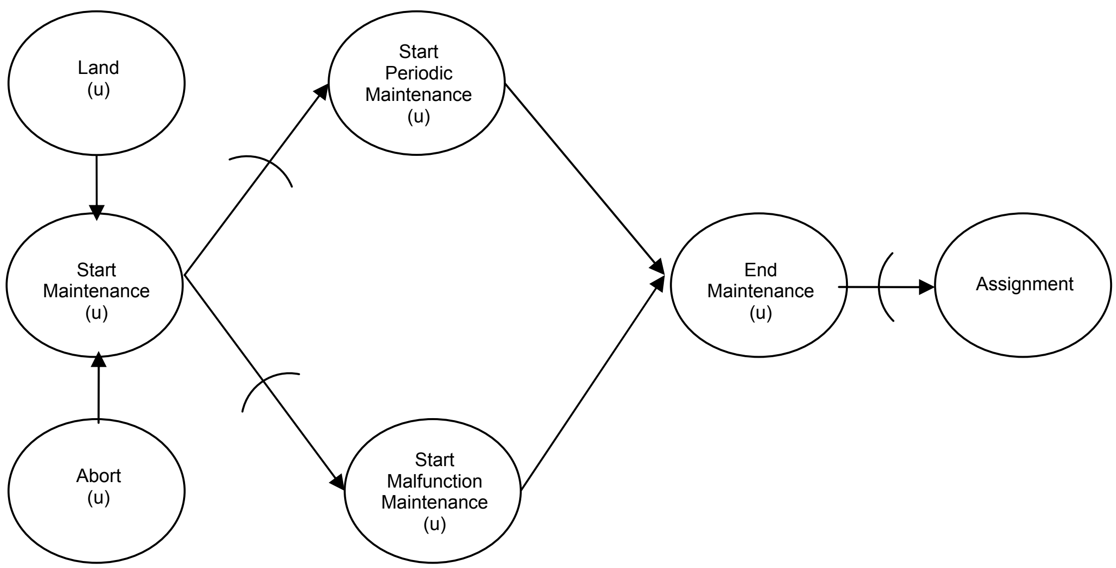


Figure 20. Event Graph for Start Maintenance

After describing the assignment problem and the entities of the execution phase, the next chapter will describe how to design the experiment in order to collect data for output analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DESIGN OF EXPERIMENTS

A. INPUT FACTORS

This section defines the input factors. Fifteen input factors are used to create the model, which will next be described in detail:

1. Coordinates of UAV Squadrons

At the beginning of the simulation run, UAVs are considered to be located at the squadrons. Locations of UAV squadrons are required for calculating the distance between the mission area and the squadrons. The user has to enter one geographic coordinate for each Heron and Gnat squadron. Since every UAV has a limited operational range, the distance between a UAV's current location and mission area must be calculated before assigning a UAV type to a mission.

2. Coordinates and Dimensions of the Operational Area

Operational area is modeled as a rectangular or square region. The user does not have to enter four different coordinates to define the shape of the operational area. Instead, only the bottom left corner of the operational area has to be defined with geographic coordinates. To define the entire area, the length and the width of the area in kilometers have to be defined by the user.

3. Mission Duration for Heron/Gnat

Mission duration is the time spent to complete one mission. It is generated by a triangular distribution with minimum, maximum and the average mission duration values. Mission duration is one of the most important factors among the performance measures. In order to be able to include mission duration into input parameters, duration was categorized as either short or long. Therefore, the user has to specify minimum, maximum and average values for each category. Mission durations also change according to the UAV type.

4. Ground Abort Rate of Heron/Gnat

This input factor defines the abort rate of Heron or Gnat before takeoff. In a simulation run, a random number is drawn for each UAV before takeoff to define if the UAV is going to abort or continue to the mission. If the generated random number is less than or equal to the predefined abort rate, the UAV aborts in preflight check and goes to maintenance. If the drawn random number is greater than the abort rate, the UAV is launched. Higher ground abort rates result in larger queues at the maintenance servers, and thus longer wait times and decreased availability for the UAVs. If a UAV aborts before takeoff, it cannot be assigned to any of the missions before the maintenance personnel fix it.

5. Crash Rate of Heron/Gnat

The simulation uses this factor to define the crash rate of Heron or Gnat during launch or landing. In a simulation run, before launching a UAV, a random number is generated. If this number is less than or equal to the drawn random number, it is assumed that the UAV crashed because of operator error or some kind of malfunction. When a UAV crashes, one UAV is subtracted from the corresponding squadron. Crashed UAVs are not replaced until the next replication of simulation.

6. Air Abort Rate of Heron/Gnat

UAVs may abort in the air because of a malfunction. Some malfunctions may only degrade the effectiveness of a mission, or effect it not at all. These types of minor malfunctions should not be included in air abort. Only the important malfunctions, which force operators to fly the UAV to the air base without completing the mission, are considered as air abort.

7. Total Number of Heron/Gnat

This factor is required to define the initial number of Herons and Gnats in the squadrons. The total number of UAVs is defined between the minimum and maximum values stated in the design points.

8. Total Number of Missions

This input factor defines the total number of missions in the mission list. In every design point, the total number of missions changes between the minimum and maximum values.

9. Total Number of Maintenance Servers for Heron/Gnat

Maintenance of UAVs can only be carried out if there is a free server. If there are two maintenance servers for Heron, that means two Herons can be served at the same time from the maintenance queue. If there are more UAVs in maintenance queue than the total number of servers, UAVs have to wait until one of the servers becomes free/available.

10. Malfunction Maintenance Time for Heron/Gnat

Every UAV has to visit the maintenance server after landing. If a UAV accomplishes all the missions and lands safely at the air base without any malfunction, it is scheduled for periodic maintenance. Periodic maintenance time is constant and predefined for both types of UAVs. If a UAV aborts on the ground or in the air, it is scheduled for malfunction maintenance. Malfunction maintenance time is a random variable with a triangular distribution.

11. Threat Level in Mission Area

While executing the missions, UAVs may encounter some kind of threats in the mission area. Surface-to-Air Missiles (SAM) or Anti-Aircraft-Artillery (AAA) may protect the targets in mission zone. These kinds of weapons threaten the UAVs. Threat level in mission areas represents the accuracy of hostile weapons that protect the target. In this simulation model, threat level can be defined as low or high. Low threat level means that the accuracy of the adversary's weapons is low. In low threat levels, UAVs have a better chance of accomplishing a mission without being shot down.

12. Weather Risk in Mission Area

Generally, operators receive hourly Meteorological Terminal Aviation Routine Weather Report (METAR) for the base and the mission area. Terminal Area Forecast (TAF) also represents some information about the next hours' weather condition. Nevertheless, these reports are never 100% accurate. Sometimes bad weather conditions may degrade mission effectiveness or sometimes may cause the crash or damaging of a UAV. That is because UAVs cannot fly in adverse weather conditions. Even if the operator gets the weather forecast; there is always a chance of encountering worse weather than expected in the mission area. Weather risk represents the severity of bad weather in the mission area. If an operator decides that the weather presents a potentially high risk for the operation, she/he may abort the mission before entering the area.

B. PERFORMANCE MEASURES

1. Mean Number of Accomplished/Unaccomplished Missions

A mission is considered to be accomplished when a UAV arrives at the mission area and stays there for the entire mission duration. The UAV has to be in the mission area at or after the Target Opportunity Window (TOW) start time and accomplish the mission before the TOW end time. The sum of the mean number of accomplished and unaccomplished missions must be equal to the total number of missions.

2. Mean Number of Crashed Herons/Gnats

UAVs can crash at takeoff or landing, or in the mission area. Operator error or important malfunctions at landing or takeoff may result in a UAV crash. In addition, UAVs may be shot down by enemy fire in the mission area. The mean number of crashed Herons/Gnats includes crashes that occur due to these three factors. The mean number of crashes for the two types of UAVs is calculated separately.

3. Mean Number of Aborted Herons/Gnats

The mean number of aborted Herons/Gnats includes ground aborts, air aborts and aborts caused by severe weather. At preflight inspection, if the maintenance team notices a malfunction, the UAV aborts. Secondly, if the predicted weather condition is worse than the UAV's acceptable risk level, the UAV also aborts. Finally, if a malfunction occurs in the air, the UAV aborts all remaining missions and returns to base for maintenance service. The mean number of aborts for the two types of UAVs is calculated separately.

4. Mean Delay Time in Maintenance Queue for Heron/Gnat

When a UAV arrives at the maintenance server and there is no free server, the UAV has to wait until the next available server. This waiting time is recorded for every UAV in each replication. Therefore, mean delay time in maintenance queue represents the mean time that Heron or Gnat spent in the maintenance queue until it starts being served by maintenance personnel.

C. DESIGN POINTS

As mentioned earlier, an NOLH design is used to set up the scenarios, since such a design provides more efficiency and flexibility compared to full factorial designs. There are 15 input factors; therefore, 129 design points are used to capture enough data from the model. All the data used to create this table were made up according to the authors' aviation background. The table below shows a part of the design points to illustrate how they are created.

Table 6. NOLH Design Points

low level	0.03	0.04	1	0.04	0.01	1	3
high level	0.2	0.12	3	0.15	0.09	3	7
decimals	3	3	0	3	3	0	0
factor name	Heron Ground Abort Rate	Heron Crash Rate	Heron Server	Gnat Ground Abort Rate	Gnat Crash Rate	Gnat Server	Total Heron
1	0.071	0.076	2	0.09	0.037	2	5
2	0.181	0.064	2	0.091	0.018	2	5
3	0.106	0.101	1	0.07	0.043	1	6
4	0.148	0.111	2	0.08	0.045	3	3
5	0.03	0.071	2	0.066	0.018	2	4
6	0.15	0.074	2	0.041	0.042	2	5
7	0.096	0.12	3	0.072	0.023	1	4
8	0.127	0.096	3	0.047	0.038	3	6
9	0.034	0.044	2	0.062	0.026	3	6
10	0.197	0.046	1	0.069	0.019	1	4
11	0.038	0.118	2	0.071	0.035	1	6
12	0.187	0.119	2	0.082	0.021	3	5
13	0.111	0.061	3	0.06	0.043	2	3
14	0.158	0.058	3	0.079	0.017	1	7
15	0.074	0.082	3	0.065	0.048	1	3
16	0.163	0.1	3	0.043	0.024	2	7
17	0.069	0.054	1	0.122	0.013	2	6
18	0.195	0.068	2	0.126	0.044	1	4
19	0.061	0.102	2	0.098	0.027	1	6
20	0.14	0.118	2	0.123	0.02	3	4
21	0.078	0.056	3	0.145	0.049	2	4

The NOLH algorithm creates the design points between the low and high levels with the decimals specified. Each row represents a design point to be used as input factors.

D. SCENARIO REPLICATION

The first step in collecting good data for output analysis is to specify the design points. After that, the number of replications is another important factor to consider. Since there are 129 fixed design points created by the NOLH algorithm,

the stochastic elements—those results that change in each replication—make the difference. Typically, a larger number of replications gives results that are more reliable. On the other hand, more replications mean more computing time to run the simulation. For this simulation experiment, 100 replications were conducted. Therefore, at each run the model creates outputs with 129 design points times 100 replications, that is, 12,900 different variations.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS AND OUTPUT ANALYSIS

This chapter will discuss the results obtained from the simulation experiments. As mentioned before, the simulation was run for 15 input factors in 129 design points with 100 replications. There are ten performance measures: accomplished missions, unaccomplished missions, crashed Herons, crashed Gnats, total aborted Herons, total aborted Gnats, total maintenance wait time for Heron, total maintenance wait time for Gnat, total assigned Herons, and total assigned Gnats. However, the high total number of missions and fewer total numbers of UAVs in the model lead all the UAVs to be used in each replication. As a result, there is no need for further analysis for total number of UAVs. First, the authors will focus on the main factors for each performance measure and then analyze the interactions, in detail, with several analysis techniques, in the following sections.

For analysis, JMP 7.0 statistical software was used. Since the input parameters are not estimated from real life data, the following results and analysis cannot be considered definitive, but can be used as a template to give insights for future analysis with real operational data.

A. MAIN FACTORS

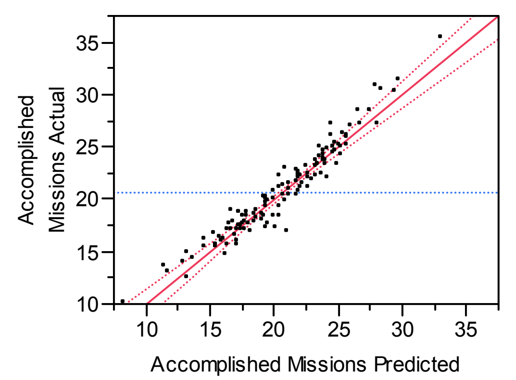
Regression analysis was used to explain the relationship between the input factors and outputs for basic analysis. While doing a regression analysis, the R^2 value, which explains how much of the variance in the data is in the model, must be checked. Then, by using the sorted parameter estimates, the authors will analyze the effect of each factor on the output. For the regression analysis, the authors formed 95% confidence interval. It is also necessary to check the residual-by-predicted plot for a random distribution of points. In other words, there should not be any pattern in this plot. The general analysis used only the main factors but not any interactions between them, in order to better comprehend the effects of each factor on the outputs.

1. Mean Number of Total Accomplished Missions

In Figure 21, it can be seen that the R^2 value is 93%. This means that the input factors of the model can explain 93% of the variance in the output. In sorted parameter estimates plot, the effects of input factors on the number of accomplished missions can be seen. According to the results, the factor with the highest impact on accomplished missions is the total number of missions. As predicted, as the number of total missions increases, the total number of accomplished missions increases. Area threat level has a negative effect on the accomplished missions. As the initial number of UAVs increases, the total number of accomplished missions increases. The increase in mission duration results in a decrease in the total number of accomplished missions since the UAVs can be assigned to fewer missions. There is a slight difference between the effects of the Heron and Gnat types due to a modeling convention that tends to select the Heron type if the bonus collected for the missions is the same for both types. In addition, the air abort rates of both UAV types and the crash rate of Heron have a negative effect on the number of accomplished missions.

Response Accomplished Missions

Actual by Predicted Plot



P<.0001 RSq=0.93 RMSE=1.2125

Summary of Fit

RSquare	0.930385
RSquare Adj	0.921144
Root Mean Square Error	1.212455
Mean of Response	20.60395
Observations (or Sum Wgts)	129

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	15	2220.0900	148.006	100.6811
Error	113	166.1153	1.470	**Prob > F**
C. Total	128	2386.2053		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Total Missions	0.3527613	0.012354	28.55	<.0001*
Area Threat Level	-3.525292	0.216027	-16.32	<.0001*
Total Herons	1.1037453	0.088242	12.51	<.0001*
Total Gnats	1.0608161	0.087602	12.11	<.0001*
Heron Mission Duration	-1.917413	0.219093	-8.75	<.0001*
Gnat Mission Duration	-1.157854	0.216494	-5.35	<.0001*
Heron Air Abort Rate	-13.66327	4.111224	-3.32	0.0012*
Gnat Air Abort Rate	-11.98423	4.115148	-2.91	0.0043*
Heron Crash Rate	-9.6946	4.640224	-2.09	0.0389*
Heron Ground Abort Rate	-3.761085	2.17256	-1.73	0.0861
Gnat Crash Rate	-6.413814	4.647451	-1.38	0.1703
Gnat Ground Abort Rate	-3.688301	3.389723	-1.09	0.2789
Heron Maintenance Servers	0.1375081	0.154405	0.89	0.3751
Weather Risk Level	0.1300774	0.221536	0.59	0.5583
Gnat Maintenance Servers	-0.040403	0.154621	-0.26	0.7943

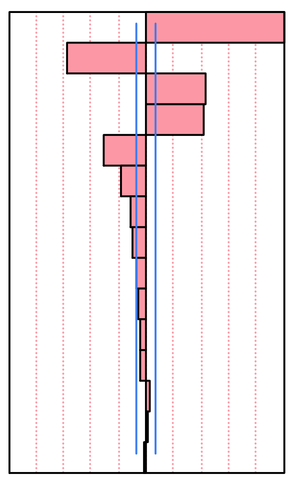


Figure 21. Regression Analysis for Accomplished Missions

Figure 22 presents the residual-by-predicted plot for accomplished missions. All points are scattered throughout the plot; the authors therefore conclude that there is no pattern.

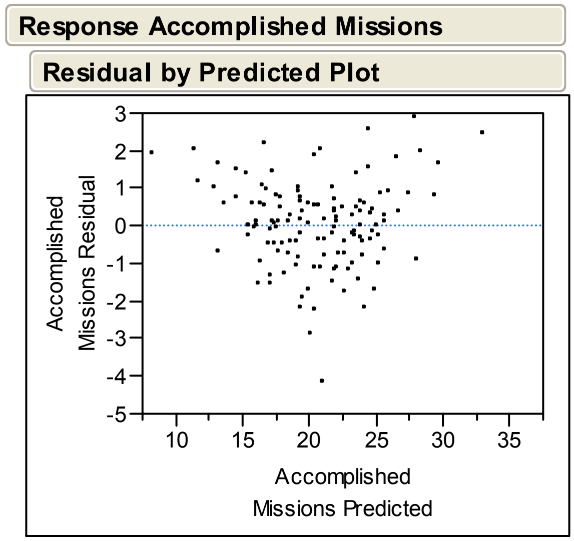


Figure 22. Residual by Predicted Plot for Accomplished Missions

2. Mean Number of Total Crashed Herons

After checking the factors that affect the mean number of accomplished missions, the authors used the same linear regression analysis technique to identify the factors that affect the mean number of crashed Herons.

In Figure 23, R^2 is approximately 0.915, which means that 91 percent of the variability is explained by the predictor variables. As the area threat level increases, the mean number of crashed Herons also increases. When the threat level of a mission area increases, more UAVs will be shot down, which will result in an increase in the mean number of crashed Herons. Another important factor seems to be the initial number of Herons. The average number of crashed Herons increases as the total number of Herons increases.

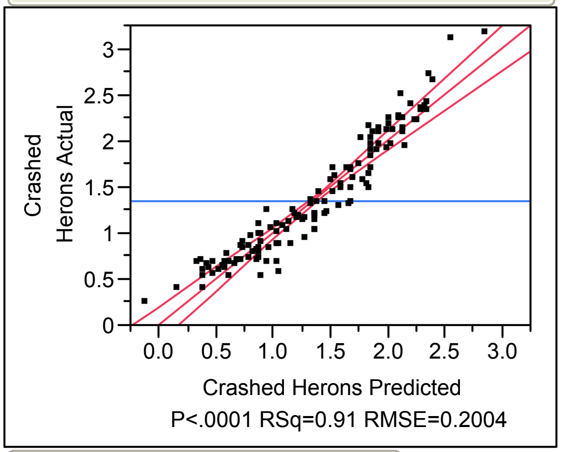
Mission duration of Heron and Gnat are also important factors that affect the mean number of crashed Herons. A decrease in Heron mission duration results in an increase in the mean number of crashed Herons. This might seem counter-intuitive at first. However, a decrease in mission duration for Herons leads to an increased number of mission assignments for Herons. Since every mission has its specific risk level, more missions mean more risk of being shot down in the mission area.

Another factor is the total number of missions. As the total number of missions increases, the mean number of crashed Herons also increases. The mission duration of Gnats also affects the mean number of crashed Herons. If the mission duration of Gnats increases, Gnats will be capable of accomplishing fewer missions because of the duration of the missions. With regard to the mission list, as Gnats accomplish fewer missions on the list, the rest of the missions will be assigned to Herons. Finally, more launched Herons will result in more crashes. It is obvious that the crash rate of Herons has a direct affect on the mean number of crashed Herons. The initial number of Gnats has a negative effect on the mean number of crashed Herons. As the initial number of Gnats decreases, the mean number of crashed Herons increases. This is because, as described before, if the initial number of Gnats decreases, more Herons will be assigned to missions and this will result in an increase in the mean number of crashed Herons.

The last factor that affects the mean number of crashed Herons is the air abort rate of Gnats. As the air abort rate of Gnats increases, the mean number of crashed Herons increases. Since there are a limited number of Gnats in the model setting, if Gnats abort in the air frequently, there will be a shortage of Gnats. Herons will be assigned to those missions that Gnats aborted and this will result in a higher number of crashed Herons.

Response Crashed Herons

Actual by Predicted Plot



Summary of Fit

RSquare	0.914938
RSquare Adj	0.903647
Root Mean Square Error	0.200387
Mean of Response	1.353876
Observations (or Sum Wgts)	129

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	15	48.806161	3.25374	81.0299
Error	113	4.537501	0.04015	**Prob > F**
C. Total	128	53.343662		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Area Threat Level	0.8996697	0.035704	25.20	<.0001*
Total Herons	0.1842178	0.014584	12.63	<.0001*
Heron Mission Duration	-0.424042	0.03621	-11.71	<.0001*
Total Missions	0.0198157	0.002042	9.71	<.0001*
Gnat Mission Duration	0.3351022	0.035781	9.37	<.0001*
Heron Crash Rate	4.6343125	0.766907	6.04	<.0001*
Total Gnats	-0.044409	0.014478	-3.07	0.0027*
Gnat Air Abort Rate	1.4297943	0.680125	2.10	0.0378*
Heron Ground Abort	-0.588242	0.359067	-1.64	0.1042
Gnat Crash Rate	0.78176	0.768101	1.02	0.3110
Heron Air Abort Rate	-0.598754	0.679477	-0.88	0.3801
Weather Risk Level	-0.03007	0.036614	-0.82	0.4132
Gnat Ground Abort	0.4553581	0.560232	0.81	0.4180
Gnat Maintenance Servers	0.0105377	0.025555	0.41	0.6809
Heron Maintenance Servers	-0.009922	0.025519	-0.39	0.6982

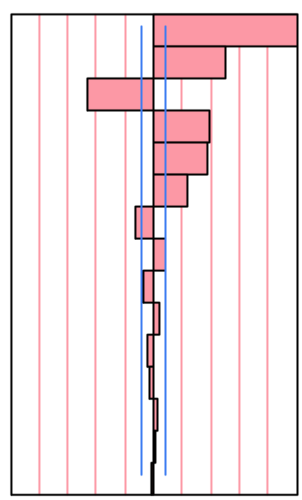


Figure 23. Regression Analysis for Crashed Herons

As before, the residual-by-predicted plot for crashed Herons was checked. As seen in Figure 24, the plot does not have any specific pattern.

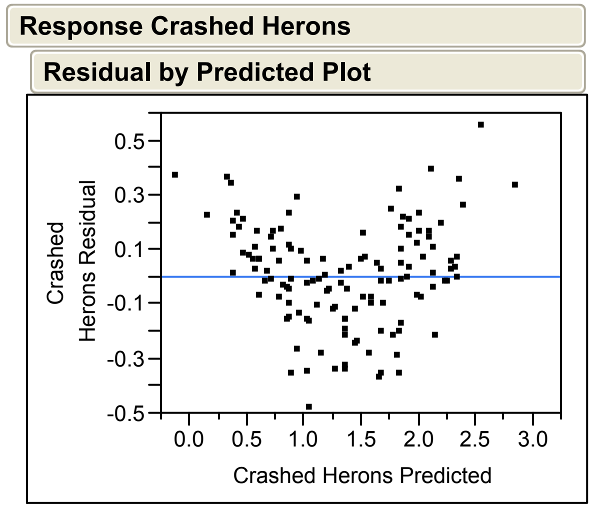


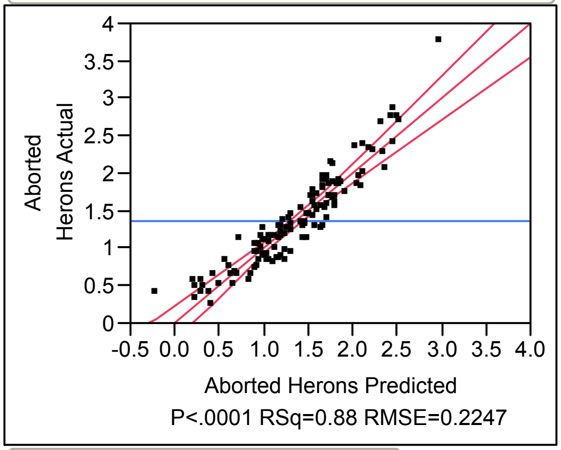
Figure 24. Residual by Predicted Plot for Crashed Herons

3. Mean Number of Total Aborted Herons

Figure 25 represents the analysis results for the mean number of aborted Herons. The ground and air abort rates for Herons and the initial number of Herons directly affect the mean number of aborted Herons. It is obvious that as these parameters increase, the mean number of aborted Herons also increases. As the mission duration of Herons or the initial number of Gnats decrease, the mean number of aborted Herons increases. When the mission duration of Herons decreases, Herons will be able to accomplish more missions in one sortie. More missions in one sortie means increased probability of aborting for UAVs. When the initial number of Gnats decreases, more Herons will be assigned to missions, which will increase the probability of Heron aborts. Finally, the mean number of Herons that are aborted increases as the total number of missions and mission durations of Gnats increase.

Response Aborted Herons

Actual by Predicted Plot



Summary of Fit

RSquare	0.883295
RSquare Adj	0.867803
Root Mean Square Error	0.224668
Mean of Response	1.363256
Observations (or Sum Wgts)	129

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	15	43.169654	2.87798	57.0168
Error	113	5.703778	0.05048	**Prob > F**
C. Total	128	48.873433		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Heron Air Abort Rate	12.467917	0.761812	16.37	<.0001*
Heron Ground Abort	6.5090406	0.402576	16.17	<.0001*
Total Herons	0.208968	0.016351	12.78	<.0001*
Heron Mission Duration	-0.39326	0.040598	-9.69	<.0001*
Total Missions	0.0218945	0.002289	9.56	<.0001*
Gnat Mission Duration	0.2950545	0.040116	7.35	<.0001*
Total Gnats	-0.042653	0.016233	-2.63	0.0098*
Heron Crash Rate	-1.059615	0.859836	-1.23	0.2204
Gnat Air Abort Rate	0.839398	0.762539	1.10	0.2733
Gnat Maintenance Servers	-0.031259	0.028651	-1.09	0.2776
Area Threat Level	-0.041092	0.04003	-1.03	0.3068
Heron Maintenance Servers	-0.022824	0.028611	-0.80	0.4267
Gnat Ground Abort	-0.438473	0.628117	-0.70	0.4866
Weather Risk Level	0.0200559	0.041051	0.49	0.6261
Gnat Crash Rate	-0.202593	0.861175	-0.24	0.8144

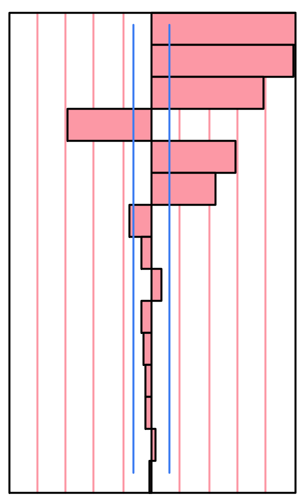


Figure 25. Regression Analysis for Aborted Herons

The residual-by-predicted plot in Figure 26 does not have any specific pattern; thus, it seems to be scattered.

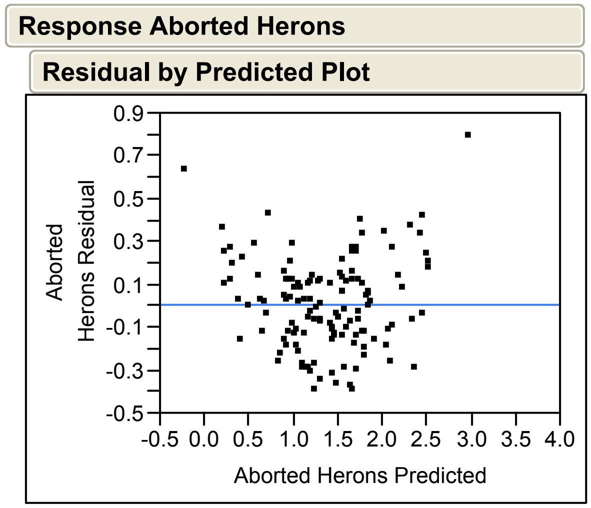


Figure 26. Residual by Predicted Plot for Aborted Herons

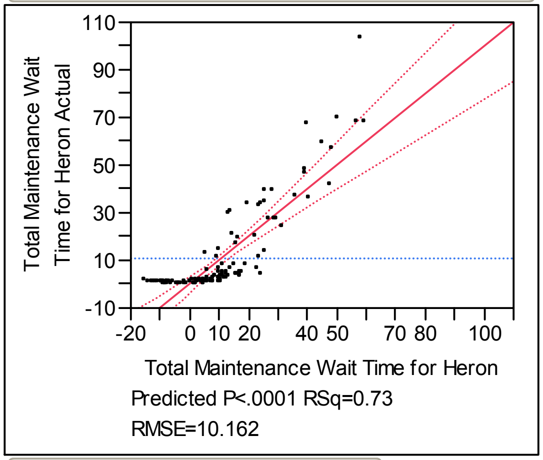
4. Mean Wait Time in Maintenance Server for Heron

First, linear regression analysis was used for the total maintenance wait time for Herons (Figure 27). When the actual-by-predicted plot was checked, results from the simulation run did not satisfy the assumptions of linear regression. From the plot, it is easy to see that most of the output data are accumulated around zero and after a certain point, there is a dramatic increase. Linear regression tries to find the best linear model that can explain the results; however, in this situation, a linear model cannot explain the results. The plot seems to be exponential rather than linear.

Only 59 percent of the variability is explained by the predictor variables in this case. The low R^2 value proves that the linear regression does not work well for explaining the mean wait time in maintenance. In addition, the fitted line extends below zero even though wait time in a server for a UAV cannot be below zero. In other words, a UAV cannot be served before it arrives to the maintenance server, so the minimum value that should be seen is zero.

Response Total Maintenance Wait Time for Heron

Actual by Predicted Plot



Summary of Fit

RSquare	0.726049
RSquare Adj	0.702832
Root Mean Square Error	10.16202
Mean of Response	10.56708
Observations (or Sum Wgts)	129

Analysis of Variance

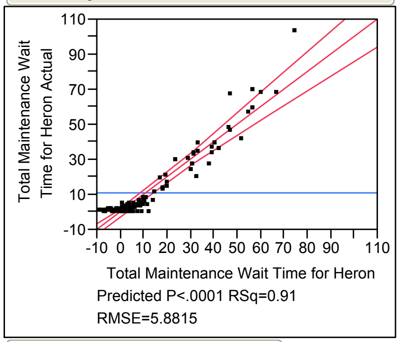
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Ratio**
Model	10	32294.917	3229.49	31.2733
Error	118	12185.467	103.27	**Prob > F**
C. Total	128	44480.384		<.0001*

Figure 27. Regression Analysis for Total Maintenance Wait Time for Heron

For such results achieved with linear regression, there are other ways to overcome the problem. One of the first methods is to try quadratic analysis. As seen in Figure 28, with quadratic analysis, R^2 increased from 59 percent to 91 percent and actual-by-predicted plot looks better. There are still some negative estimates for waiting time in server, but the sorted parameter estimates will provide information about the most important factors affecting the output.

Response Total Maintenance Wait Time for Heron

Actual by Predicted Plot



Summary of Fit

RSquare	0.913678
RSquare Adj	0.900457
Root Mean Square Error	5.881453
Mean of Response	10.56708
Observations (or Sum Wgts)	129

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	17	40640.729	2390.63	69.1104
Error	111	3839.655	34.59	**Prob > F**
C. Total	128	44480.384		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Heron Maintenance Servers	-18.25306	0.744651	-24.51	<.0001*
(Heron Maintenance Servers-2)*(Heron Maintenance Servers-2)	16.221038	1.11337	14.57	<.0001*
(Heron Maintenance Servers-2)*(Total Herons-5)	-6.133432	0.603635	-10.16	<.0001*
Total Herons	3.6650824	0.423085	8.66	<.0001*
(Heron Maintenance Servers-2)*(Heron Air Abort Rate-0.055)	-204.0539	32.99168	-6.19	<.0001*
Heron Air Abort Rate	86.708879	19.86266	4.37	<.0001*
(Heron Maintenance Servers-2)*(Total Missions-35)	-0.383597	0.089674	-4.28	<.0001*
Heron Ground Abort	42.971629	10.48719	4.10	<.0001*
Total Missions	0.1848033	0.059488	3.11	0.0024*
(Heron Crash Rate-0.07997)*(Heron Maintenance Servers-2)	88.675001	36.00997	2.46	0.0153*
(Heron Ground Abort-0.11501)*(Heron Maintenance Servers-2)	-46.49651	19.72673	-2.36	0.0202*
(Heron Maintenance Servers-2)*(Gnat Ground Abort-0.095)	51.130986	22.08968	2.31	0.0225*
Gnat Ground Abort	-34.65542	16.21477	-2.14	0.0348*
(Heron Maintenance Servers-2)*(Area Treat Level-0.49612)	3.2233629	1.509036	2.14	0.0349*
Area Treat Level	-1.922107	1.044393	-1.84	0.0684
(Total Herons-5)*(Heron Air Abort Rate-0.055)	32.825674	19.68826	1.67	0.0983
Heron Crash Rate	-12.95209	22.33896	-0.58	0.5632

Figure 28. Regression Analysis for Total Maintenance Wait Time for Heron

Quadratic analysis seems to perform better than linear regression; however, they are both unable to explain the exponential increase in the wait time. For this situation, it will be better to use an approximation formula for the total mean wait time and try to determine the factors causing this exponential increase.

$$W = \left(\frac{CV_a^2 + CV_s^2}{2} \right) \left(\frac{u}{1-u} \right) t_s,$$

In the formula above, *W* is the mean wait time in the maintenance queue. CV_a stands for the coefficient of variation for the arrival process and CV_s stands for the coefficient of variation for the maintenance service process. A positive correlation indicates a higher likelihood of further arrivals; conversely, a negative correlation will indicate the opposite. t_s stands for the mean maintenance time. Since the arrival processes and the maintenance times are following the same distributions throughout the whole simulation and are not the cause of the exponential increase, the only factor that will cause such an increase in the wait time seems to be the utilization of the servers. In the formula, (u/1-u) represents the impact of the server utilization on the wait time in the maintenance queue. In this case, when there is a sufficient number of servers (three servers), the mean wait time stays around zero. When the number of available servers decreases to one, the mean wait time increases exponentially as the slack in capacity disappears (Yücesan, 2007). The effects of the factors will be discussed in more detail in the key interactions section.

B. KEY INTERACTIONS

The previous section only considered the impact of the main factors on the output of the model. While the main factors are the ones that have the most important effects on performance, it is also crucial to analyze at least some of the interactions of these factors to better understand the behavior of the model. There are different approaches for analyzing the interactions. Multiple regression analysis, stepwise regression analysis and partition tree are the major ones. It will be good to see all the interactions of the factors, but in reality for 15 factors, it is nearly impossible to analyze meaningfully all of the interactions. Even though it can be accomplished, interpreting an interaction of the 14^{th} degree is a tough problem. Therefore, with such a high number of factors, it is more practical to

conduct a stepwise regression analysis by first eliminating the unimportant factors and interactions and then building the analysis model with the remaining important ones.

In the following section, stepwise regression analysis with two-way interactions is made for each performance measures. Confidence interval is set to 0.95 in JMP 7.0 to eliminate unimportant factors and interactions. After identifying the important factors, a multiple regression analysis model is constructed to analyze the factors and their interactions in detail.

As explained in the main factors section, only the factors about one UAV type will be analyzed. The analysis for the other type will be made in a similar manner.

1. Mean Number of Total Accomplished Missions

In order to conduct the detailed output analysis for accomplished missions, a stepwise regression analysis was first run with the main factors and their two-way interactions. Figure 29 is the step history for the stepwise regression model, showing which interactions are significant enough to enter into the multiple regression analysis model. As a result, the model is constructed with 15 main factors plus 21 interactions.

After identifying the most important factors that affect the number of accomplished missions, the authors created a multiple regression model with these 36 inputs. The most important thing to point out here is the new R^2 value. By adding the interactions into the model, it is expected that the R^2 value will increase; if not, then there is no value in adding the interactions into the model other than making the model more complex and harder to explain.

Stepwise Fit

Response: Accomplished Missions

Stepwise Regression Control

Prob to Enter 0.050
Prob to Leave 0.050
Direction: Forw ard
Rules: Combine

1 row s not used due to excluded row s or missing values.

Step History

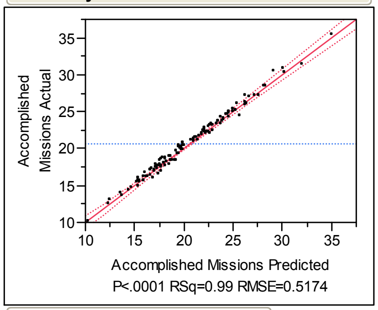
Step	Parameter	Action	"Sig Prob"	Seq SS	RSquare	Cp	p
1	(Total Missions-35)*(Area Threat Level-0.49612)	Entered	0.0000	1587.478	0.6653	2655	4
2	(Total Herons-5)*(Total Gnats-5)	Entered	0.0000	447.3702	0.8528	1106.2	7
3	(Heron Mission Duration-0.49612)*(Gnat Mission Duration-0.49612)	Entered	0.0000	186.1612	0.9308	465.14	10
4	(Total Missions-35)*(Heron Mission Duration-0.49612)	Entered	0.0000	29.53128	0.9431	364.51	11
5	(Gnat Air Abort Rate-0.055)*(Gnat Mission Duration-0.49612)	Entered	0.0000	35.25379	0.9579	245.98	13
6	(Heron Crash Rate-0.07997)*(Heron Air Abort Rate-0.055)	Entered	0.0000	22.64083	0.9674	173.29	16
7	(Heron Ground Abort Rate-0.11501)*(Total Herons-5)	Entered	0.0001	11.97854	0.9724	135.66	18
8	(Heron Ground Abort Rate-0.11501)*(Heron Crash Rate-0.07997)	Entered	0.0011	6.138273	0.9750	116.33	19
9	(Heron Maintenance Servers-2)*(Heron Air Abort Rate-0.055)	Entered	0.0012	6.951893	0.9779	96.166	21
10	(Gnat Crash Rate-0.05001)*(Heron Mission Duration-0.49612)	Entered	0.0030	5.472579	0.9802	81.146	23
11	(Gnat Air Abort Rate-0.055)*(Weather Risk Level-0.49612)	Entered	0.0142	3.712865	0.9818	72.242	25
12	(Total Herons-5)*(Heron Mission Duration-0.49612)	Entered	0.0164	2.375736	0.9828	65.985	26
13	(Heron Maintenance Servers-2)*(Gnat Crash Rate-0.05001)	Entered	0.0114	2.512135	0.9838	59.254	27
14	(Heron Ground Abort Rate-0.11501)*(Gnat Ground Abort Rate-0.095)	Entered	0.0296	2.6258	0.9849	54.128	29
15	(Total Gnats-5)*(Heron Mission Duration-0.49612)	Entered	0.0178	1.995523	0.9857	49.193	30
16	(Total Gnats-5)*(Heron Air Abort Rate-0.055)	Entered	0.0136	2.057894	0.9866	44.04	31
17	(Heron Ground Abort Rate-0.11501)*(Gnat Maintenance Servers-2)	Entered	0.0345	2.163502	0.9875	40.521	33
18	(Total Missions-35)*(Gnat Mission Duration-0.49612)	Entered	0.0436	1.256325	0.9880	38.155	34
19	(Heron Maintenance Servers-2)*(Weather Risk Level-0.49612)	Entered	0.0480	1.16891	0.9885	36.092	35
20	(Gnat Maintenance Servers-2)*(Total Gnats-5)	Entered	0.0368	1.259453	0.9891	33.715	36
21	(Gnat Crash Rate-0.05001)*(Gnat Maintenance Servers-2)	Entered	0.0211	1.474387	0.9897	30.591	37

Figure 29. Step History Diagram for Stepwise Regression of Accomplished Missions

In the main factors section, the value of R^2 is calculated as 93%. Figure 30 shows that the new R^2 value is 99%, which means that by adding 21 interactions, the input factors can explain nearly all the variability for accomplished missions. Also, it can be seen in the sorted parameter estimates that some interactions have larger t-ratios, meaning that they are more significant than some of the main effects in explaining the output.

Response Accomplished Missions

Actual by Predicted Plot



Summary of Fit

RSquare	0.98968
RSquare Adj	0.985641
Root Mean Square Error	0.517379
Mean of Response	20.60395
Observations (or Sum Wgts)	129

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	36	2361.5786	65.5994	245.0652
Error	92	24.6267	0.2677	**Prob > F**
C. Total	128	2386.2053		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Total Missions	0.3539248	0.005272	67.13	<.0001*
Area Threat Level	-3.526476	0.092187	-38.25	<.0001*
Total Herons	1.1043027	0.037656	29.33	<.0001*
Total Gnats	1.0598186	0.037384	28.35	<.0001*
Heron Mission Duration	-1.916382	0.093492	-20.50	<.0001*
Gnat Mission Duration	-1.155732	0.092383	-12.51	<.0001*
(Total Missions-35)*(Heron Mission Duration-0.49612)	-0.136024	0.011303	-12.03	<.0001*
(Total Missions-35)*(Area Threat Level-0.49612)	-0.139787	0.011637	-12.01	<.0001*
(Gnat Air Abort Rate-0.055)*(Gnat Mission Duration-0.49612)	31.305596	3.916734	7.99	<.0001*
Heron Air Abort Rate	-13.67343	1.754346	-7.79	<.0001*
Gnat Air Abort Rate	-12.15952	1.75617	-6.92	<.0001*
(Heron Maintenance Servers-2)*(Heron Air Abort Rate-0.055)	-18.08831	3.361922	-5.38	<.0001*
Heron Crash Rate	-9.69868	1.980092	-4.90	<.0001*
(Total Herons-5)*(Total Gnats-5)	-0.182586	0.038454	-4.75	<.0001*
(Heron Ground Abort Rate-0.11501)*(Total Herons-5)	-3.999451	0.922158	-4.34	<.0001*
(Gnat Air Abort Rate-0.055)*(Weather Risk Level-0.49612)	15.400717	3.728822	4.13	<.0001*
Heron Ground Abort Rate	-3.759613	0.927077	-4.06	0.0001*
(Heron Ground Abort Rate-0.11501)*(Gnat Maintenance Servers-2)	-5.740792	1.545961	-3.71	0.0004*
(Total Gnats-5)*(Heron Mission Duration-0.49612)	0.3342547	0.095801	3.49	0.0007*
(Heron Crash Rate-0.07997)*(Heron Air Abort Rate-0.055)	412.60897	121.4204	3.40	0.0010*
Gnat Crash Rate	-6.407474	1.983267	-3.23	0.0017*
(Heron Ground Abort Rate-0.11501)*(Gnat Ground Abort Rate-0.095)	-111.7873	36.29933	-3.08	0.0027*
(Heron Maintenance Servers-2)*(Weather Risk Level-0.49612)	-0.43949	0.145935	-3.01	0.0034*
(Gnat Maintenance Servers-2)*(Total Gnats-5)	0.2003848	0.069761	2.87	0.0051*
(Total Gnats-5)*(Heron Air Abort Rate-0.055)	-5.085761	1.877326	-2.71	0.0080*
(Heron Ground Abort Rate-0.11501)*(Heron Crash Rate-0.07997)	-127.2921	49.82135	-2.55	0.0123*
Gnat Ground Abort Rate	-3.675565	1.44647	-2.54	0.0127*
(Gnat Crash Rate-0.05001)*(Gnat Maintenance Servers-2)	-11.73059	4.998308	-2.35	0.0211*
(Gnat Crash Rate-0.05001)*(Heron Mission Duration-0.49612)	11.231779	4.805572	2.34	0.0216*
(Total Missions-35)*(Gnat Mission Duration-0.49612)	-0.034591	0.015455	-2.24	0.0276*
Heron Maintenance Servers	0.1389836	0.06589	2.11	0.0376*
(Heron Mission Duration-0.49612)*(Gnat Mission Duration-0.49612)	-0.443798	0.215168	-2.06	0.0420*
(Total Herons-5)*(Heron Mission Duration-0.49612)	-0.159011	0.081446	-1.95	0.0539
(Heron Maintenance Servers-2)*(Gnat Crash Rate-0.05001)	6.2320241	3.282067	1.90	0.0607
Weather Risk Level	0.1306577	0.094538	1.38	0.1703
Gnat Maintenance Servers	-0.040039	0.06598	-0.61	0.5455

Figure 30. Multiple Regression Analysis for Accomplished Missions

The effects of the main factors were already discussed in the previous section. This section will focus more on the effects of the interactions on the performance measures. Interaction profiles can clearly show the interaction between the inputs and their effects on changing situations. Figure 31 shows some of the noteworthy interaction profiles.

For example, it can be seen that as the total number of Gnats increases from three to seven, the number of accomplished missions also increases for both settings with three and seven Herons. However, there is a slight difference between two settings. If there are three Herons, the number of accomplished missions will increase to around 20. However, if there are seven Herons, the number of accomplished missions increases to around 25.

There is another interesting result for the interaction between the air abort rate of Heron and the number of Heron maintenance servers. When the number of maintenance servers is one, the number of accomplished missions for both air abort rates of 0.1 and 0.01 is the same. Nevertheless, as the number of maintenance servers increases to three, the air abort rate of Heron makes a difference on the number of accomplished missions. For an abort rate of 0.1, the number of accomplished missions decreases and for 0.01 the number of accomplished missions increases.

Analysis of the interaction between total number of missions and area threat level shows how important the total number of missions is to the output. When the area threat level is minor, there is a huge difference in the number of accomplished missions compared to the numbers of total missions. For 20 missions, the number of accomplished missions is more than 15 while for 50 missions the number of accomplished missions is around 30. However, for 20 initial missions, as the threat level of the area increases, the number of accomplished missions stays nearly the same, but for 50 initial missions it decreases dramatically to the 20s. Actually, this is not a surprising result; the

decreased ratios of total accomplished missions are nearly the same for both situations, but the initial number of missions shows the changes to be more significant.

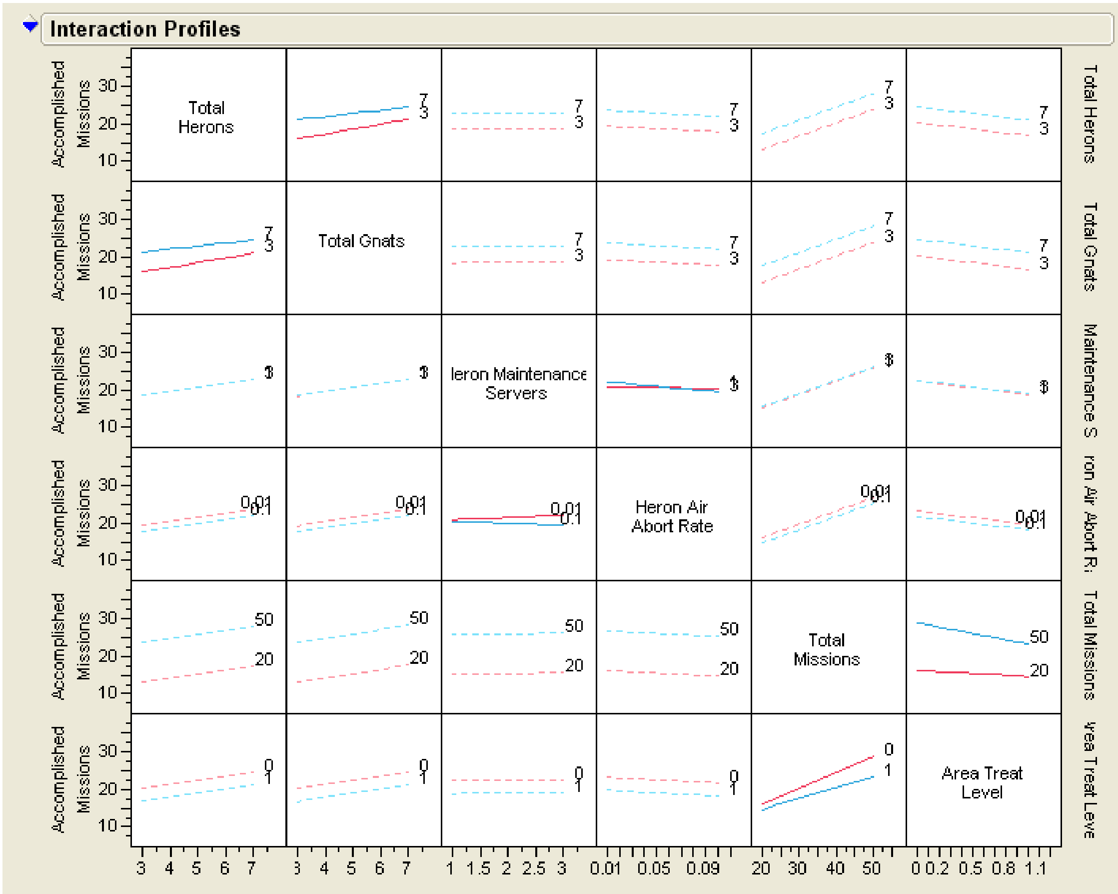


Figure 31. Interaction Profiles for Accomplished Missions

2. Mean Number of Total Crashed Herons

After conducting a stepwise regression analysis, 22 two-way interactions other than the main factors (shown in Figure 32) are added as inputs to the multiple regression model. Again, the purpose is to explain the variability of the number of total crashed Herons in a better way.

Stepwise Fit

Response: Crashed Herons

Stepwise Regression Control

Prob to Enter 0.050
Prob to Leave 0.050
Direction: Forw ard
Rules: Combine

1 row s not used due to excluded row s or missing values.

Step History

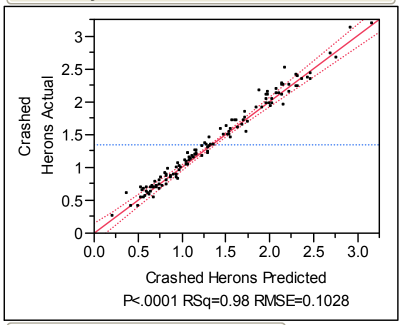
Step	Parameter	Action	"Sig Prob"	Seq SS	RSquare	Cp	p
1	(Total Herons-5)*(Area Threat Level-0.49612)	Entered	0.0000	35.16611	0.6592	1389.6	4
2	(Total Missions-35)*(Heron Mission Duration-0.49612)	Entered	0.0000	8.44803	0.8176	693.52	7
3	(Heron Crash Rate-0.07997)*(Gnat Mission Duration-0.49612)	Entered	0.0000	4.947556	0.9104	288.38	10
4	(Total Missions-35)*(Area Threat Level-0.49612)	Entered	0.0003	0.511863	0.9200	247.85	11
5	(Heron Mission Duration-0.49612)*(Area Threat Level-0.49612)	Entered	0.0001	0.5445	0.9302	204.6	12
6	(Gnat Mission Duration-0.49612)*(Area Threat Level-0.49612)	Entered	0.0003	0.395065	0.9376	173.77	13
7	(Total Gnats-5)*(Gnat Air Abort Rate-0.055)	Entered	0.0002	0.52932	0.9475	135.78	16
8	(Heron Air Abort Rate-0.055)*(Area Threat Level-0.49612)	Entered	0.0004	0.367617	0.9544	109.23	18
9	(Total Herons-5)*(Total Missions-35)	Entered	0.0015	0.212761	0.9584	93.551	19
10	(Heron Ground Abort Rate-0.11501)*(Total Herons-5)	Entered	0.0048	0.209233	0.9623	80.164	21
11	(Heron Ground Abort Rate-0.11501)*(Heron Crash Rate-0.07997)	Entered	0.0151	0.108519	0.9643	73.146	22
12	(Total Herons-5)*(Heron Air Abort Rate-0.055)	Entered	0.0161	0.101642	0.9662	66.7	23
13	(Heron Mission Duration-0.49612)*(Gnat Mission Duration-0.49612)	Entered	0.0134	0.102465	0.9682	60.185	24
14	(Gnat Maintenance Servers-2)*(Heron Air Abort Rate-0.055)	Entered	0.0267	0.115377	0.9703	54.597	26
15	(Total Missions-35)*(Weather Risk Level-0.49612)	Entered	0.0274	0.108907	0.9724	49.547	28
16	(Gnat Ground Abort Rate-0.095)*(Heron Mission Duration-0.49612)	Entered	0.0334	0.09783	0.9742	45.417	30
17	(Total Herons-5)*(Heron Mission Duration-0.49612)	Entered	0.0401	0.058229	0.9753	42.579	31
18	(Total Gnats-5)*(Gnat Mission Duration-0.49612)	Entered	0.0072	0.095076	0.9771	36.678	32
19	(Total Herons-5)*(Gnat Air Abort Rate-0.055)	Entered	0.0209	0.066465	0.9783	33.154	33
20	(Gnat Crash Rate-0.05001)*(Weather Risk Level-0.49612)	Entered	0.0373	0.078196	0.9798	30.656	35
21	(Gnat Maintenance Servers-2)*(Total Herons-5)	Entered	0.0211	0.060305	0.9809	27.645	36
22	(Total Herons-5)*(Gnat Mission Duration-0.49612)	Entered	0.0383	0.046652	0.9818	25.768	37

Figure 32. Step History Diagram for Stepwise Regression of Crashed Herons

After creating the multiple regression model with 37 inputs, the R^2 value increased to 98%. Figure 33 shows the parameter estimates and their effects on the total number of crashed Herons. In the actual-by-predicted plot, the dots are closer to the 45˚ line. This means that the model can predict the actual number of crashed Herons in a very effective way. As mentioned above, since most factors and their effects are interpreted in the main factors section, this section will focus on the effects of important interactions.

Response Crashed Herons

Actual by Predicted Plot



Summary of Fit

RSquare	0.98178
RSquare Adj	0.97465
Root Mean Square Error	0.102784
Mean of Response	1.353876
Observations (or Sum Wgts)	129

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	36	52.371720	1.45477	137.7025
Error	92	0.971942	0.01056	**Prob > F**
C. Total	128	53.343662		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Area Threat Level	0.9010247	0.018257	49.35	<.0001*
Total Herons	0.1840413	0.007481	24.60	<.0001*
Heron Mission Duration	-0.422932	0.018574	-22.77	<.0001*
Total Missions	0.0198539	0.001046	18.98	<.0001*
Gnat Mission Duration	0.3331806	0.018346	18.16	<.0001*
Heron Crash Rate	4.6125654	0.391673	11.78	<.0001*
(Heron Mission Duration-0.49612)*(Area Threat Level-0.49612)	-0.376995	0.047144	-8.00	<.0001*
(Gnat Mission Duration-0.49612)*(Area Threat Level-0.49612)	0.3016496	0.03916	7.70	<.0001*
(Total Herons-5)*(Area Threat Level-0.49612)	0.1260071	0.018414	6.84	<.0001*
Total Gnats	-0.044748	0.007427	-6.03	<.0001*
(Total Herons-5)*(Total Missions-35)	0.0059693	0.001019	5.86	<.0001*
(Total Missions-35)*(Area Threat Level-0.49612)	0.0135233	0.002332	5.80	<.0001*
(Heron Ground Abort Rate-0.11501)*(Total Herons-5)	-0.767574	0.185899	-4.13	<.0001*
Gnat Air Abort Rate	1.4277817	0.348843	4.09	<.0001*
(Total Herons-5)*(Heron Mission Duration-0.49612)	-0.101449	0.02479	-4.09	<.0001*
(Heron Air Abort Rate-0.055)*(Area Threat Level-0.49612)	-2.894285	0.747388	-3.87	0.0002*
(Total Gnats-5)*(Gnat Mission Duration-0.49612)	0.0789106	0.0217	3.64	0.0005*
Heron Ground Abort Rate	-0.582466	0.184023	-3.17	0.0021*
(Total Missions-35)*(Weather Risk Level-0.49612)	-0.007481	0.002424	-3.09	0.0027*
(Total Herons-5)*(Heron Air Abort Rate-0.055)	0.9951851	0.350762	2.84	0.0056*
(Gnat Maintenance Servers-2)*(Total Herons-5)	0.0416963	0.014999	2.78	0.0066*
(Heron Ground Abort Rate-0.11501)*(Heron Crash Rate-0.07997)	-23.22246	8.710297	-2.67	0.0091*
(Gnat Crash Rate-0.05001)*(Weather Risk Level-0.49612)	2.5024812	0.970606	2.58	0.0115*
(Heron Mission Duration-0.49612)*(Gnat Mission Duration-0.49612)	0.0952474	0.03823	2.49	0.0145*
(Gnat Ground Abort Rate-0.095)*(Heron Mission Duration-0.49612)	-1.537148	0.623067	-2.47	0.0155*
(Total Herons-5)*(Gnat Mission Duration-0.49612)	0.034755	0.016539	2.10	0.0383*
Gnat Crash Rate	0.7983666	0.392989	2.03	0.0451*
(Total Missions-35)*(Heron Mission Duration-0.49612)	-0.005735	0.003005	-1.91	0.0594
(Total Herons-5)*(Gnat Air Abort Rate-0.055)	0.6707308	0.356579	1.88	0.0631
(Gnat Maintenance Servers-2)*(Heron Air Abort Rate-0.055)	-0.981909	0.522632	-1.88	0.0634
Heron Air Abort Rate	-0.600191	0.348112	-1.72	0.0880
Gnat Ground Abort Rate	0.4481724	0.286787	1.56	0.1215
Weather Risk Level	-0.029244	0.018748	-1.56	0.1222
(Total Gnats-5)*(Gnat Air Abort Rate-0.055)	0.4163847	0.362235	1.15	0.2533
Gnat Maintenance Servers	0.0100917	0.013097	0.77	0.4429
(Heron Crash Rate-0.07997)*(Gnat Mission Duration-0.49612)	-0.379888	0.872273	-0.44	0.6642

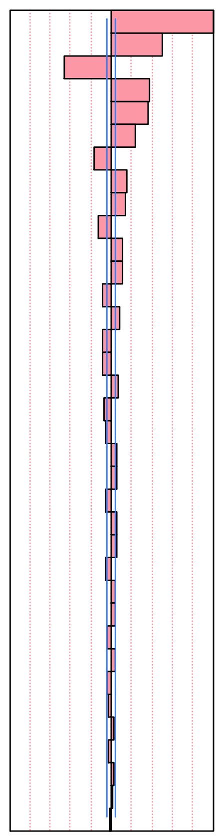


Figure 33. Multiple Regression Analysis for Crashed Herons

In Figure 34, the interaction profiles plot shows some of the interactions that have significant effects on the number of crashed Herons. When the interaction between ground abort rate of Heron and total number of Herons is analyzed, it can be seen that as the initial number of Herons was three, the ground abort rate of Herons is not affecting the number of crashed Herons. As the total number of Herons increases to seven, the ground abort rate starts to make a difference. If the ground abort rate is higher, the number of crashed Herons decreases. This is logical because, as the ground aborts increase they cause fewer Herons to take off for missions; thus, fewer casualties occur.

Another important interaction is between the total number of Herons and missions. When there is a small number of missions, there is just a small amount of difference in the number of crashed Herons. For three Herons, the number of casualties is around one while for seven Herons, the average number of casualties is around 1.5. However, as the number of missions increases, the number of Herons has a greater importance in the number of casualties.

The third interaction to be analyzed is between the total number of Herons and air abort rate of Heron. When the initial number of Herons is seven, the air abort rate of Herons has no effect on the number of crashed Herons. However, for three initial Herons, as the air abort rate increases, the number of casualties decreases because higher air abort rates imply that fewer Herons continue their missions.

The last interaction affecting the number of crashed Herons is between the total number of Herons and the area threat level. Normally, area threat level is the most important factor in the number of casualties. Nevertheless, its effect also changes with the initial number of Herons. While there is less threat, the initial number of Herons is not as effective as in the higher threat levels. In the high threat situation, as the initial number of Herons increases, the number of crashed Herons increases more rapidly.

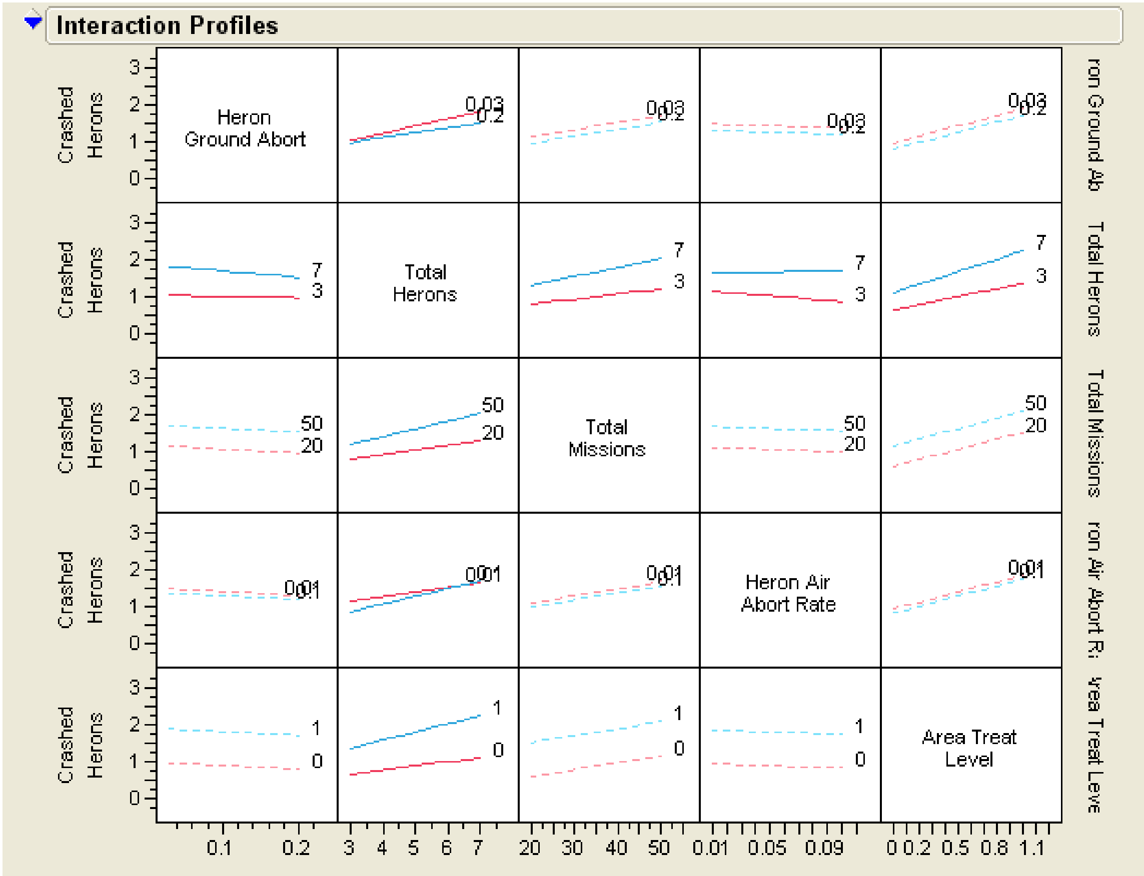


Figure 34. Interaction Profiles for Crashed Herons

3. Mean Number of Total Aborted Herons

The results of stepwise regression analysis for aborted Herons are shown in Figure 35. According to the results, 22 two-way interactions are sufficiently significant to be used with the main factors for detailed analysis. The list is not sorted according to the importance of the interactions; this means that it will be incorrect to take the first couple of interactions to do the analysis with the most important interactions. If the concern is to analyze just a couple of the most important interactions, then the probability to enter for the stepwise regression analysis can be set to smaller probabilities (such as 0.001) to capture fewer but more important interactions.

Stepwise Fit

Response: Total Aborted Herons

Stepwise Regression Control

Prob to Enter 0.050

Prob to Leave 0.050

Direction: Forw ard

Rules: Combine

1 row s not used due to excluded row s or missing values.

Step History

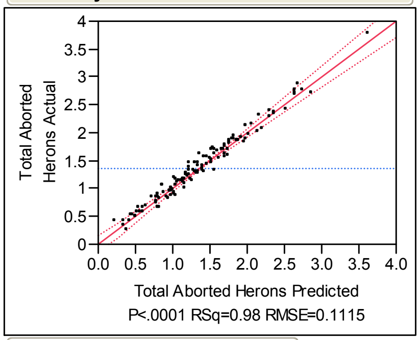
Step	Parameter	Action	"Sig Prob"	Seq SS	RSquare	Cp	p
1	(Heron Ground Abort Rate-0.11501)*(Heron Air Abort Rate-0.055)	Entered	0.0000	23.51163	0.4811	1361.9	4
2	(Total Herons-5)*(Total Missions-35)	Entered	0.0000	12.16302	0.7299	656.73	7
3	(Heron Mission Duration-0.49612)*(Gnat Mission Duration-0.49612)	Entered	0.0000	7.157454	0.8764	244.24	10
4	(Heron Air Abort Rate-0.055)*(Heron Mission Duration-0.49612)	Entered	0.0003	0.64594	0.8896	208.47	11
5	(Total Herons-5)*(Heron Air Abort Rate-0.055)	Entered	0.0001	0.66429	0.9032	171.63	12
6	(Heron Air Abort Rate-0.055)*(Gnat Mission Duration-0.49612)	Entered	0.0001	0.576696	0.9150	139.91	13
7	(Total Missions-35)*(Heron Air Abort Rate-0.055)	Entered	0.0002	0.489485	0.9250	113.29	14
8	(Heron Ground Abort Rate-0.11501)*(Total Herons-5)	Entered	0.0001	0.447964	0.9342	89.095	15
9	Total Gnats	Entered	0.0003	0.356991	0.9415	70.222	16
10	(Total Herons-5)*(Gnat Mission Duration-0.49612)	Entered	0.0014	0.249536	0.9466	57.632	17
11	(Heron Ground Abort Rate-0.11501)*(Heron Crash Rate-0.07997)	Entered	0.0030	0.261901	0.9519	46.318	19
12	(Heron Crash Rate-0.07997)*(Gnat Mission Duration-0.49612)	Entered	0.0064	0.155751	0.9551	39.212	20
13	(Heron Ground Abort Rate-0.11501)*(Area Threat Level-0.49612)	Entered	0.0132	0.170508	0.9586	33.242	22
14	(Heron Maintenance Servers-2)*(Total Missions-35)	Entered	0.0109	0.166701	0.9620	27.495	24
15	(Gnat Maintenance Servers-2)*(Gnat Air Abort Rate-0.055)	Entered	0.0207	0.168544	0.9655	23.64	27
16	(Total Missions-35)*(Heron Mission Duration-0.49612)	Entered	0.0174	0.092262	0.9674	20.246	28
17	(Heron Maintenance Servers-2)*(Area Threat Level-0.49612)	Entered	0.0303	0.073499	0.9689	17.948	29
18	(Heron Ground Abort Rate-0.11501)*(Gnat Maintenance Servers-2)	Entered	0.0265	0.074189	0.9704	15.61	30
19	(Heron Crash Rate-0.07997)*(Total Missions-35)	Entered	0.0219	0.075865	0.9719	13.175	31
20	(Heron Maintenance Servers-2)*(Heron Mission Duration-0.49612)	Entered	0.0367	0.060634	0.9732	11.629	32
21	(Heron Crash Rate-0.07997)*(Gnat Air Abort Rate-0.055)	Entered	0.0192	0.073183	0.9747	9.3503	33
22	(Heron Mission Duration-0.49612)*(Area Threat Level-0.49612)	Entered	0.0366	0.055913	0.9758	8.081	34

Figure 35. Step History Diagram for Stepwise Regression of Aborted Herons

As mentioned for the previous analysis, the first thing to check is the R^2 value to evaluate the explanatory power of interaction. The new R^2 value in Figure 36 is 98%, which proves the importance of the interactions when compared with the R^2 value of the model with just the main factors. In addition, the actual-by-predicted plot is aligned well. Normally, the most important factors are the ground and air abort rates of Herons. There are also important interactions that affect the performance metrics more than some of the main factors.

Response Total Aborted Herons

Actual by Predicted Plot



Summary of Fit

RSquare	0.975826
RSquare Adj	0.967428
Root Mean Square Error	0.11152
Mean of Response	1.363256
Observations (or Sum Wgts)	129

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	33	47.691953	1.44521	116.2060
Error	95	1.181479	0.01244	**Prob > F**
C. Total	128	48.873433		<.0001*

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Heron Air Abort Rate	12.470517	0.37808	32.98	<.0001*
Heron Ground Abort Rate	6.5045487	0.199807	32.55	<.0001*
Total Herons	0.2083356	0.008074	25.80	<.0001*
Heron Mission Duration	-0.391727	0.020044	-19.54	<.0001*
Total Missions	0.0218644	0.001135	19.26	<.0001*
Gnat Mission Duration	0.29423	0.019789	14.87	<.0001*
(Heron Air Abort Rate-0.055)*(Heron Mission Duration-0.49612)	-7.415985	0.896505	-8.27	<.0001*
(Total Herons-5)*(Heron Air Abort Rate-0.055)	2.7479966	0.365034	7.53	<.0001*
(Heron Ground Abort Rate-0.11501)*(Total Herons-5)	1.7735957	0.259326	6.84	<.0001*
(Heron Air Abort Rate-0.055)*(Gnat Mission Duration-0.49612)	4.9461012	0.827831	5.97	<.0001*
Total Gnats	-0.043267	0.00804	-5.38	<.0001*
(Total Missions-35)*(Heron Air Abort Rate-0.055)	0.2380628	0.050402	4.72	<.0001*
(Heron Ground Abort Rate-0.11501)*(Heron Crash Rate-0.07997)	-30.91992	8.589464	-3.60	0.0005*
(Heron Crash Rate-0.07997)*(Gnat Mission Duration-0.49612)	3.2357852	0.90229	3.59	0.0005*
(Total Herons-5)*(Gnat Mission Duration-0.49612)	0.0616031	0.017372	3.55	0.0006*
(Heron Ground Abort Rate-0.11501)*(Gnat Maintenance Servers-2)	-1.135923	0.359982	-3.16	0.0021*
(Total Herons-5)*(Total Missions-35)	0.0030243	0.000971	3.11	0.0024*
(Heron Maintenance Servers-2)*(Total Missions-35)	0.0059125	0.002003	2.95	0.0040*
(Heron Mission Duration-0.49612)*(Gnat Mission Duration-0.49612)	0.1197965	0.041841	2.86	0.0052*
(Total Missions-35)*(Heron Mission Duration-0.49612)	-0.006687	0.002485	-2.69	0.0084*
(Heron Crash Rate-0.07997)*(Gnat Air Abort Rate-0.055)	-66.81112	24.93906	-2.68	0.0087*
(Heron Ground Abort Rate-0.11501)*(Area Threat Level-0.49612)	1.1592214	0.436161	2.66	0.0092*
(Heron Crash Rate-0.07997)*(Total Missions-35)	-0.137191	0.055512	-2.47	0.0152*
(Heron Maintenance Servers-2)*(Heron Mission Duration-0.49612)	-0.073533	0.03051	-2.41	0.0179*
Heron Crash Rate	-1.015179	0.424255	-2.39	0.0187*
(Heron Maintenance Servers-2)*(Area Threat Level-0.49612)	-0.072983	0.030981	-2.36	0.0205*
Gnat Air Abort Rate	0.8800769	0.376426	2.34	0.0215*
Gnat Maintenance Servers	-0.030668	0.014119	-2.17	0.0323*
(Heron Mission Duration-0.49612)*(Area Threat Level-0.49612)	0.0915216	0.043164	2.12	0.0366*
Area Threat Level	-0.040883	0.019868	-2.06	0.0424*
(Gnat Maintenance Servers-2)*(Gnat Air Abort Rate-0.055)	1.2661498	0.694854	1.82	0.0716
Heron Maintenance Servers	-0.023621	0.014117	-1.67	0.0976
(Heron Ground Abort Rate-0.11501)*(Heron Air Abort Rate-0.055)	2.205868	8.298028	0.27	0.7909

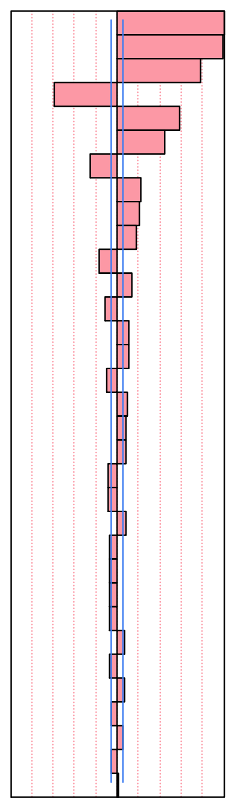


Figure 36. Multiple Regression Analysis for Aborted Herons

The important interactions can be seen in the interaction profiles plot in Figure 37. The authors will analyze four of the interactions for detailed analysis of aborted Herons, starting with the interaction of ground abort rate of Heron and the total number of Herons. When there are three Herons, the ground abort rate does not have as big an impact as it does when there are seven initial Herons. As the number of Herons increases to seven, low abort rates do not affect the number of total aborts very much, while higher abort rates increase total aborts more rapidly.

There is a similar result for the interaction between the initial number of Herons and the air abort rate of Herons. For smaller air abort rates, the number of aborted Herons is nearly the same for three and seven initial Herons. As the air abort rate increases, a huge difference emerges between the numbers of initial Herons and the aborted ones. For low rates, the number of aborted Herons is around one for both situations. As the air abort rate increases, the number of aborted herons for two situations increases differently. However, this difference is just in numbers and the increase ratio is nearly the same.

Another important interaction can be seen for air abort rate and mission durations for Herons. For low air abort rates, the number of aborted Herons is the same for low and high mission durations. Low mission durations lead to more total aborts as the air abort rate increases. This is because the aborts occur while the UAVs ingress from one mission area to another. The shorter times that the UAV spends in a mission area means that it spends more time to ingress and is more vulnerable to abort.

Finally, the interaction between the crash rate and the ground abort rate of Herons was analyzed. For low ground abort rates, the crash rate does not make a big difference in the total number of aborts. However, as the abort rate increases, the crash rate and the number of total aborts change inversely. As more Herons crash, there are fewer Herons susceptible to being aborted during mission.

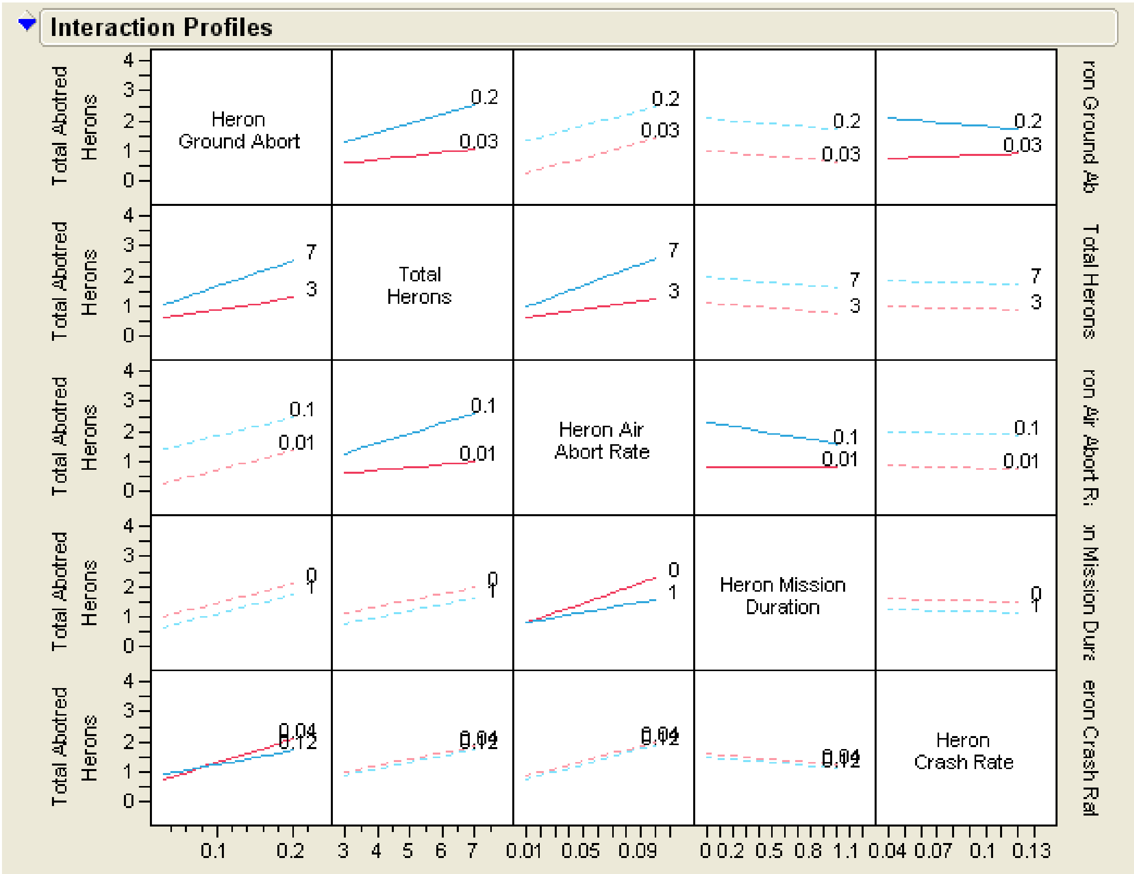


Figure 37. Interaction Profiles for Aborted Herons

4. Mean Wait Time in Maintenance Server for Heron

As explained in the main factors section, linear and quadratic regression analyses are not good enough to explain the total maintenance wait time for Heron. Therefore, a partition tree was used to find out the important factors that are affecting the waiting time of Herons in the maintenance queue. Figure 38 is the partition tree for three splits, showing the most important factors affecting the total maintenance wait times. Since the inputs are not collected from real life data, the wait times seem to be too small. However, in reality there will be situations where the wait times increase dramatically to hours. Therefore, the concern of the analysis about the total maintenance wait time for Heron is to find out the factors to minimize this delay time.

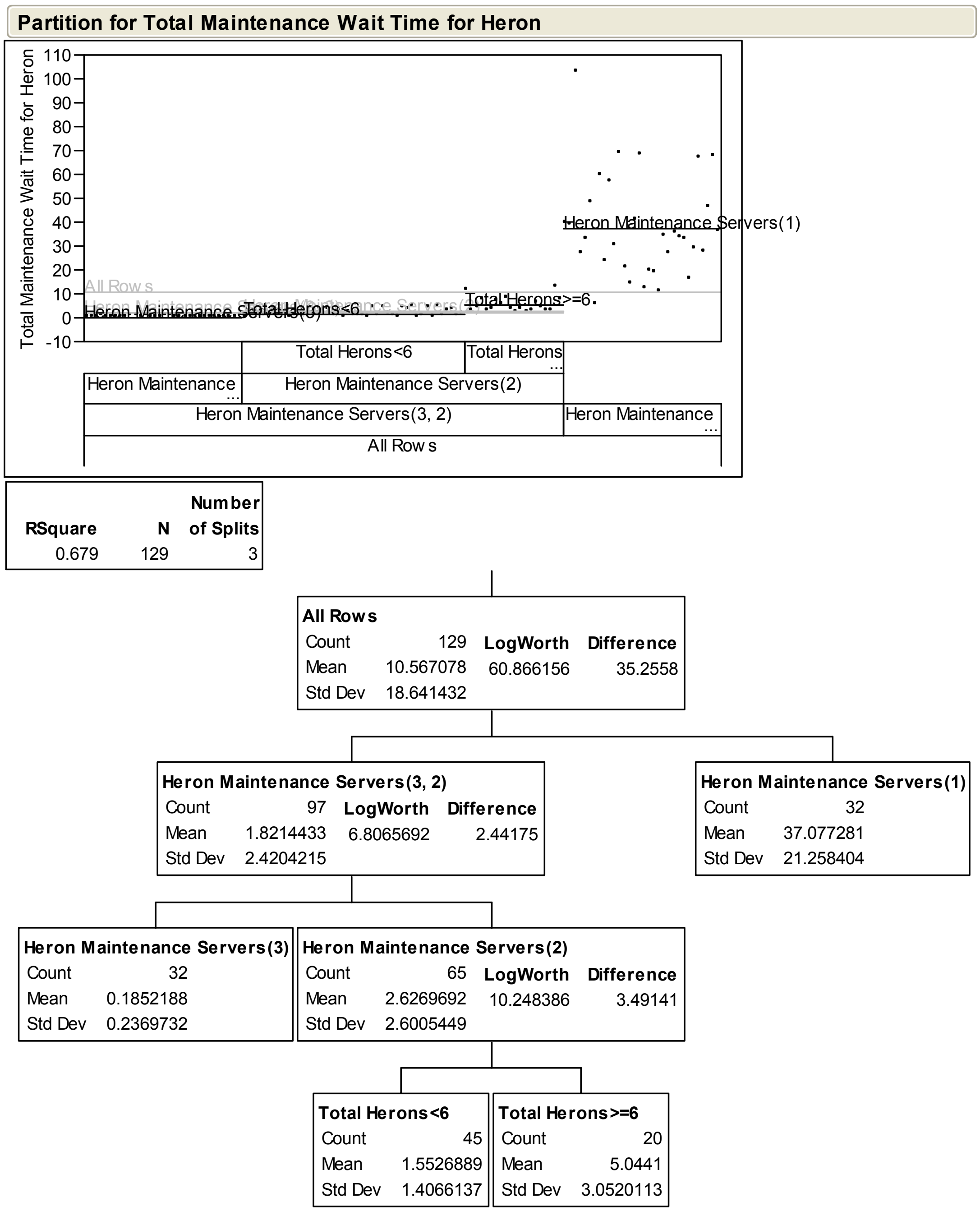


Figure 38. Partition tree for maintenance wait time for Heron

The mean wait time for Heron is 10.567 minutes with a standard deviation of 18.641 minutes. Since the main concern is to reduce the wait time, it is more important to find the minimum wait time than the biggest R^2. In the first split, a

huge difference can be seen between the mean wait times on the right and the left sides of the tree. The most important factor impacting the wait time is the number of maintenance servers. If there are two or three servers, the wait time drops to 1.82 minutes and the standard deviation is 2.42 minutes. If there is only one server, then the mean wait time is 37.07 minutes and the standard deviation is 21.26 minutes.

Since the goal is to find ways to reduce the wait time, the tree is split from the left side. If there are three servers, the wait time drops to 0.18 minutes with a standard deviation of 0.24 minutes. If there are only two servers, the mean wait time is 2.63 minutes with a standard deviation of 2.6 minutes. In the third split, it can be seen that there are two servers, the important factor is whether there are six or more Herons. If there are less than six Herons while having two maintenance servers, the mean wait time is 1.55 minutes and the standard deviation is 1.4 minutes. If there are more than six Herons then the mean wait time jumps up to 5.04 minutes with a standard deviation of 3.05 minutes.

As a conclusion, in order to find ways to reduce the maintenance wait time for Heron, there must be three maintenance servers. If that is not possible, it is better to have two servers with less than six Herons. The R^2 value of 67% will seem to be less but even three splits provided the main factors affecting the total maintenance wait time for Herons. Therefore, there is no need to make further partitions to get a higher R^2 value.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

A. SUMMARY

In this thesis, an assignment and simulation tool is created for assigning UAVs to missions in an optimal and robust way in situations where the decision makers possess all the information on the next day’s missions. There are two main phases run in this tool. The first phase solves the mission assignment problem with a nearly optimal solution. For this phase, the authors tried to include as many factors as possible that affect the problem. Some factors considered not to be important have not been included in many earlier studies; the include such things as ground abort rates of the UAVs or the changing weather conditions. After solving the assignment problem, the second phase was to evaluate the robustness of this algorithm with a stochastic simulation created for this project. To accomplish this, a full UAV operation cycle has been simulated, starting from the assignment phase, going through all the preflight activities, traveling to the mission area, conducting mission and post-mission activities, and finally ending with the return of the UAV to the base and accomplishment of the maintenance to make it ready for the next missions.

In the analysis phase, using NOLH, the authors examined all the parameters and their interactions that affect the results. The results proved that some parameter that seemed not to be important would be as important as the other parameters. The overall results proved that the algorithm works effectively and creates plausible results.

Since no data were collected from actual UAV operations, the authors cannot claim that the results will provide insights about real-life situations. However, the goal of this thesis is to create a template model that can be modified with real-life data. By changing the input factors, the decision makers can use this model to help them make decisions about UAV assignment problems for real operations. The model will also be used for other purposes,

such as deciding the UAV demand if the possible missions are known. In addition, the maintenance issues will be observed and possible solutions created by analyzing the maintenance queue wait times and other factors.

B. FUTURE WORK

Great effort was made to include as many factors as possible into the model to create a more realistic simulation model. However, many more factors can be inserted into the model. An analyst with real-life data can run the model with these parameters and conduct a more realistic output analysis to be used by the decision makers. Second, the assignment model is one of the key contributions of this thesis. There is no one optimum solution to solve the assignment problem. Therefore, another future work will be to create different assignment problem solutions and compare them with each other to see the advantages and disadvantages. There are many constraints on UAV operation in real life. The number of ground control stations, their abilities to control UAVs, or personnel constraints might significantly influence the results, so these kinds of issues can be added into the model to analyze their effects. Logistics is another important issue for all military operations; for example, the lack of a part in the logistics flow would create maintenance issues and could affect the whole operation's success. Therefore, logistics is another area that should be incorporated into the model. Another future work will be extending the duration of the simulation to more than one day. Since this study was conducted for a one-day scenario, there will be different constraints and results for multiple day scenarios. The last recommendation for future work is to add a Graphical User Interface (GUI) to the model to make it more understandable by non-programmers.

LIST OF REFERENCES

Anderson, P. S. (2002). Development of a UAV ground control station. (Master of Science, Linkoping University).

Army Unmanned Aircraft System Operations. (2006). Retrieved April 2008, from *www.**uav**forum.com/library/**army**_fmi3-04-155.pdf*.

Buss, A. (1995). A tutorial on discrete-event modeling with simulation graphs. *WSC '95: Proceedings of the 27th Conference on Winter Simulation,* Arlington, VA, 74-81.

Buss, A. (1996). Modeling with event graphs. *WSC '96: Proceedings of the 28th Conference on Winter Simulation,* Coronado, CA, 153-160.

Buss, A. (2001). Discrete event programming with Simkit. *Simulation News Europe, 32*(33), 15-16.

Buss, A. (Fall 2007). *Mv4302 Advanced Discrete Event Simulation Modeling Class Notes.*

David, A. B. (2005). *IAF to acquire Heron UAV.* Retrieved April 2008, from http://www.janes.com.

Geer, H., & Bolkcon, C. (2005). *Unmanned aerial vehicles: Background and issues for Congress.* Retrieved March 2008, from www.fas.org/irp/crs/RL31872.pdf.

Gnat 750. Retrieved March 2008, from http://www.uavforum.com/vehicles/production/gnat750.htm.

IAI Heron 1. (2007). Retrieved April 2008, from http://www.janes.com.

Jane's (2007). *GA-ASI Gnat.* Retrieved April 2008, from http://www.janes.com.

Law, M. A. (2007). *Simulation modeling and analysis* (4th Ed. ed.). New York: McGraw Hill.

Pike, J. (1999). *General Atomic GNAT-750 lofty view.* Retrieved March 2008, from http://www.fas.org/irp/program/collect/gnat-750.htm.

Pike, J. (2007). *Unmanned aerial vehicles.* Retrieved March 2008, from http://www.fas.org/irp/program/collect/uav.htm.

Radharamanan, R., & Choi, L. I. (1986). A branch and bound algorithm for the traveling salesman and the transportation routing problems. *Comput. Ind. Eng. 11*(1-4), 236-240.

Sanchez, S. M. (2006). Work smarter, not harder: Guidelines for designing simulation experiments. *WSC '06: Proceedings of the 38th Conference on Winter Simulation,* Monterey, CA, 47-57.

Schriber, T. J., & Brunner, D. T. (2005). Inside discrete-event simulation software: How it works and why it matters. *1997 Winter Simulation Conference (WSC'97),* Atlanta, Georgia, 14-22.

Schruben, L. (1983). Simulation modeling with event graphs. *Communications of the ACM, 26*(11), 957-963.

Yucesan, E. (2007). *Competitive supply chains: A value-based management perspective*. U.K.: Palgrave McMillan.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Hava Kuvvetleri Komutanligi
   BILKARDES Sube Mudurlugu
   Bakanliklar - 06100
   Ankara, TURKEY

4. Hava Harp Okulu Kutuphanesi
   Yesilyurt
   Istanbul, TURKEY

5. Kara Harp Okulu Kutuphanesi
   Bakanliklar - 06100
   Ankara, TURKEY

6. Deniz Harp Okulu Kutuphanesi
   Tuzla
   Istanbul, TURKEY

7. Arnold H. Buss, Ph.D.
   MOVES Institute
   Naval Postgraduate School
   Monterey, California

8. Enver Yücesan, Ph.D.
   Technology and Operations Management Area
   INSEAD
   Boulevard de Constance
   77305 Fontainebleau Cedex
   France

9. Yücel Alver
   Hava Kuvvetleri Komutanligi
   BILKARDES Sube Mudurlugu
   Bakanliklar - 06100
   Ankara, TURKEY

10. Murat Özdoğan
8. Ana Jet Us, 182. Filo K.ligi
Diyarbakir, TURKEY